Geschäftsprozesse erfolgreich digitalisieren

Die digitale Transformation durch die Low-Code-Plattform X4 BPMS beschleunigen

X4 ESB

SoftProject

Die in dieser Dokumentation enthaltenen Informationen und die zugehörigen Programme können ohne besondere Ankündigung geändert werden. Für etwaige Fehler übernimmt SoftProject keine Haftung.

Diese Dokumentation und die zugehörigen Programme dürfen ohne schriftliche Zustimmung der SoftProject GmbH weder ganz noch teilweise kopiert, reproduziert, verändert oder in irgendeine elektronische oder maschinenlesbare Form umgewandelt werden.

Alle genannten Warenzeichen sind Warenzeichen der jeweiligen Eigentümer.

Kontakt

SoftProject GmbH

Am Erlengraben 3

D-76275 Ettlingen

Website: www.softproject.de

Vertrieb

Telefon: +49 7243 56175-0

vertrieb@softproject.de

SoftProject-Support

Telefon: +49 7243 56175-333

support@softproject.de

© SoftProject GmbH. Alle Rechte vorbehalten.

Inhaltsverzeichnis

1 Einführung 14
1.1 Was ist ein ESB Project? 14
1.2 Wie kann ich ESB Projects nutzen? 14
2 Oberfläche 15
2.1 Symbolleiste 16
2.2 Kontextmenü 17
2.3 Perspektiven 20
2.3.1 Perspektive wechseln 20
2.3.2 Perspektive zurücksetzen 20
2.4 Sichten 21
2.4.1 Sichten öffnen 28
2.5 Editoren 28
2.5.1 BPMN Editor 29
2.5.2 Datenbank-Editor 29
2.5.3 JSON-Editor 33
2.5.4 Mapping Editor 34
2.5.5 Process Editor 39
2.5.6 XML-Editor 42
3 Prozessbausteine und ihre Verwendung 44
3.1 Die Palette 44
3.2 Prozessbausteine verbinden 45
3.3 Bausteine im Bereich „Control“ 46
3.3.1 Stop 46
3.3.2 Fork und Join 47
3.3.3 Merge 49
3.3.4 Condition 50
3.3.5 Connection 64
3.3.6 Pause 65

3.3.7 Counter 67
3.3.8 Transaction 69
3.3.9 Variable 70
3.4 Bausteine im Bereich „Modules“ 73
3.4.1 Function Adapter 73
3.4.2 XML 76
3.4.3 JSON 78
3.4.4 Document 81
3.4.5 Subprocess 93
3.4.6 Mapping 95
3.4.7 PDF Creator 105
3.5 Elemente im Bereich „Drawing“ 107
3.5.1 Connector 107
3.5.2 Rectangle und Ellipse 108
3.5.3 Textbox 109
3.5.4 Polygon 110
3.5.5 Polyline und Line 111
3.5.6 Image 111
4 Mit ESB-Projekten arbeiten 113
4.1 Templates 113
4.2 Die Struktur von ESB-Projekten 113
4.3 ESB-Projekt anlegen 113
4.4 Projekte verwalten 114
4.4.1 Repository-Elemente importieren 115
4.4.2 Repository-Elemente exportieren 116
4.4.3 Repository-Elemente intelligent umbenennen 116
4.4.4 Repository-Elemente intelligent verschieben 118
4.4.5 Repository-Referenzen reparieren 119
4.4.6 Nicht verwendete Ressourcen löschen 120
4.4.7 Repository-Elemente durchsuchen 120
4.4.8 Veraltete Projekte migrieren 122

5 Prozesse anlegen und modellieren 123
5.1 Prozess anlegen 123
5.1.1 Nutzung von Private Files unterhalb von Prozessen 124
5.2 Eigenschaften eines Technical Process (.wrf) 125
5.3 Prozessbausteine ausrichten 127
5.4 Prozesse validieren 128
5.4.1 Prozess-Validierung durchführen 128
5.4.2 Standardmäßig hinterlegte Validierungsregeln 129
5.4.3 Benutzerdefinierte Validierungsregeln 130
5.5 Prozessbausteine parametrieren 132
5.5.1 Möglichkeiten der Parametrierung 132
5.5.2 Anwendungsbeispiele für Parameter 136
5.5.3 Dynamische Parameter verwenden 137
5.5.4 Component-Parameter verwenden 140
5.5.5 Status mit ConditionState dynamisch setzen 142
5.5.6 Parameter in XSL-Mappings verwenden 143
5.6 Parameter erstellen und verwenden 146
5.6.1 System-Parameter 146
5.6.2 Globale Parameter und -Gruppen 152
5.6.3 Projekt-Parameter 164
5.7 XSL-Hilfsfunktionen 183
5.7.1 TimeHelper (Veraltet) 183
5.7.2 WorkingTimeCalculator (Veraltet) 193
5.7.3 checkXML 199
5.8 XSLT-Funktionen 200
5.9 XPath-Funktionen 200
5.9.1 Knotengruppen-Funktionen (node-set) 201
5.9.2 Boolesche Funktionen (boolean) 201
5.9.3 Number-Funktionen 202
5.9.4 Zeichenketten-Funktionen (String) 203
5.10 Entwickler-Hilfswerkzeuge 205

5.11 URL-Protokolle....207
5.11.1 context:-URL-Protokoll....207
5.11.2 xstore://-URL-Protokoll....208
5.11.3 x4db:/-URL-Protokoll....209
5.11.4 cryptstore://-URL-Protokoll....210
5.11.5 globalcontext:-URL-Protokoll....212
5.12 Logging von Technical Processes....213
5.12.1 .log-Datei aktivieren....214
5.13 Save Points für wiederaufsetzbare Prozesse verwenden....215
5.14 Webservices via HTTP(S) bereitstellen....216
5.14.1 Webservice-Konfiguration grafisch bearbeiten....218
5.14.2 ReSTful-Webservices bereitstellen....221
5.14.3 SOAP-Webservices bereitstellen....225
5.14.4 File-Webservice bereitstellen....240
5.15 Webservices über Open API bereitstellen....242
5.15.1 Webservice-Konfiguration grafisch bearbeiten....243
5.15.2 ReSTful Webservices bereitstellen....245
5.15.3 Technical Process als OpenAPI-ReST-Service bereitstellen....246
5.16 COBOL Copybook Converter Wizard....250
5.16.1 Voraussetzungen....251
5.16.2 Funktionen im COBOL Copybook Wizard....251
5.17 SessionID einer Web App auslesen....252
5.18 Gruppen eines Benutzers auslesen....252
5.19 Claims eines Benutzers auslesen....252
6 Prozesse debuggen und ausführen....253
6.1 Technical Process debuggen....253
6.1.1 Debug-Session ohne Debugging ausführen....255
6.1.2 Debug-Session beenden....255
6.1.3 Debugger mit Server verbinden....255
6.2 Prozess ausführen....256
6.3 Statuswerte für Prozesse und Prozess-Schritte....256

7 Scheduler 259
7.1 Scheduler erstellen 259
7.2 Prozesse zeitgesteuert ausführen 260
7.2.1 Neue zeitgesteuerte Ausführung anlegen 260
7.2.2 Serverstart 260
7.2.3 Cron 261
8 Prozessmonitoring 263
8.1 Monitoring von Profiling-Daten in der Design-Ansicht konfigurieren 263
8.2 Monitoring von Profiling-Daten in der Source-Ansicht konfigurieren 265
8.2.1 Beispiel einer .profiling-Datei 266
9 Mit Adapter-Projekten arbeiten 267
10 Übersicht der X4 Adapter 268
10.1 Über X4 Adapter 268
10.2 Big Data 268
10.2.1 Amazon S3 DynamoDB Connector 268
10.2.2 Apache Cassandra Connector 279
10.2.3 Google Cloud Storage BigTable Connector 287
10.2.4 Microsoft Azure Data Lake Gen 1 Storage Connector 293
10.3 Cloud 300
10.3.1 Amazon S3 Simple Storage Connector 300
10.3.2 Amazon SimpleDB Connector 310
10.3.3 Dropbox Connector 316
10.3.4 Google Cloud Storage Objects Connector 320
10.3.5 Google Drive Connector 330
10.3.6 Microsoft Azure Blob Storage Connector 335
10.3.7 Microsoft Azure Table Storage Connector 342
10.4 Communication 348
10.4.1 AS/400 PCML 348
10.4.2 File System 350
10.4.3 File System Bulk 362
10.4.4 FTP 370

10.4.5 iCalendar 389
10.4.6 Ping 395
10.4.7 SCP 396
10.4.8 TCP/IP Connector/Listener 399
10.4.9 Telnet Connector 400
10.5 Compression 403
10.5.1 gzip Converter 403
10.5.2 ZIP 404
10.6 Datenbanken 409
10.6.1 JDBC Connector 409
10.6.2 JDBC Iterator 433
10.6.3 JDBC Metadata Explorer 454
10.7 EDI 466
10.7.1 EDIFACT Adapter 466
10.7.2 Unterstützte EDIFACT-Nachrichten-Typen 474
10.8 E-Mail 479
10.8.1 IMAP 479
10.8.2 MIME Multipart 485
10.8.3 MIME Multipart Reader 507
10.8.4 POP/SMTP 509
10.9 ERP 516
10.9.1 Abas ERP Connector 516
10.9.2 SAP Connector 522
10.10 File Processing 576
10.10.1 ASCII to XML 577
10.10.2 Base64 Batch Decoder 583
10.10.3 Base64 Converter 584
10.10.4 Charset Changer 585
10.10.5 CSV 586
10.10.6 Encoding Converter 601
10.10.7 Hex Converter 602

10.10.8 Image Processing 603
10.10.9 Java Properties File Reader 607
10.10.10 QR Code Generator 608
10.10.11 Regular Expression 610
10.10.12 Stream Splitter 611
10.10.13 Structured Text 612
10.10.14 Text Replacement 615
10.10.15 Transliteration Adapter 616
10.11 Geo Location 617
10.11.1 Bing Maps 618
10.11.2 Geo Calculation 623
10.11.3 Google Maps 631
10.12 Insurance 637
10.12.1 Audatex DV90 Converter 637
10.12.2 GDV 638
10.12.3 SZF Reader 643
10.13 Internet of Things 644
10.13.1 IEC 61850 Client 644
10.13.2 Modbus TCP Connector 648
10.13.3 MQTT Publisher/Subscriber 651
10.13.4 OPC Client Connector 666
10.13.5 OPC UA Publisher/Subscriber 669
10.13.6 SIMATIC S7 Connector 682
10.14 Message Oriented Middleware 684
10.14.1 Apache Kafka 684
10.14.2 AMQP Publisher/Subscriber 696
10.14.3 JMS 703
10.14.4 Websphere MQ 710
10.15 Office 736
10.15.1 Microsoft Excel 736
10.15.2 Microsoft Word Properties 748

10.15.3 Microsoft Word to Text Converter 750
10.15.4 Working Time Calculator 751
10.16 PDF Processing 755
10.16.1 PDF Data Processing 755
10.16.2 PDF Image Processing 771
10.16.3 PDF Split and Merge 781
10.17 Security 787
10.17.1 Antivirus 788
10.17.2 BCrypt Calculator 790
10.17.3 Checksum Creator 792
10.17.4 Digest Calculator 793
10.17.5 JSON WebToken Calculator 794
10.17.6 LDAP Connector 798
10.17.7 Password Generator 805
10.17.8 Random Number Generator 806
10.17.9 RSA Encryption 808
10.17.10 SAML Calculator 809
10.17.11 SSH Connector 813
10.17.12 TOTP One-Time Password Generator 815
10.17.13 UUID Generator 818
10.17.14 XML Encryption 819
10.17.15 XML Signature 825
10.17.16 AES Encryption 826
10.18 System Administration 833
10.18.1 JMeter Test Execution 834
10.18.2 JMX MBean 839
10.18.3 JMX Number 844
10.18.4 JMX Switch 846
10.18.5 SNMP Management 847
10.18.6 Windows Event Reader 851
10.18.7 Windows Services Explorer 853

10.19 Tools 854
10.19.1 WebSMS 855
10.20 Web Services 856
10.20.1 JSON 856
10.20.2 OData Connector 861
10.20.3 URL Encoding Converter 872
10.20.4 URL Reader 874
10.20.5 Webservice 875
10.20.6 X4 Streaming Support 895
10.20.7 XOP (MTOM) 897
10.20.8 YAML Converter 908
10.21 X4 API 910
10.21.1 Deployments 910
10.21.2 Global Parameter Management 915
10.21.3 Keycloak Management 918
10.21.4 Process Monitoring 978
10.21.5 Process Monitoring Management 990
10.21.6 Process Runtime Management 996
10.21.7 Project Parameter Management 998
10.21.8 Savepoint Management 1005
10.21.9 Schedule Management 1011
10.21.10 System Configuration Management 1017
10.21.11 Third Party License Summary 1021
10.21.12 Web App Configuration Management 1022
10.21.13 Web App Theme Management 1026
10.22 X4 Tools 1033
10.22.1 Case Management Documents 1033
10.22.2 Case Management 1036
10.22.3 Case Management Statistics 1039
10.22.4 Commandline 1043
10.22.5 Document Comparison 1046

10.22.6 Document Properties Changer 1048
10.22.7 Empty Output Adapter 1049
10.22.8 Global Context Manipulation 1050
10.22.9 Keycloak Token 1052
10.22.10 Log4J Logging 1053
10.22.11 Loop Adapter 1055
10.22.12 Parallel Process Starter 1056
10.22.13 Process Starter 1058
10.22.14 Projects 1061
10.22.15 Project Access 1063
10.22.16 System Status 1073
10.22.17 Task Management 1079
10.22.18 User Identity Adapter 1082
10.22.19 User Session Adapter 1086
10.22.20 Web App Translation 1087
10.22.21 X4 Management 1088
10.22.22 X4 Server Config Reader 1089
10.22.23 X4 Variable Collector 1091
10.23 XML Tools 1093
10.23.1 Inline Content Expander 1094
10.23.2 Record Counter 1097
10.23.3 XHTML Form Extractor 1098
10.23.4 XML Checker 1103
10.23.5 XML Element Wrapper 1103
10.23.6 XML Formatter 1104
10.23.7 XML Iterator 1105
10.23.8 XML Text Extractor 1106
10.23.9 XML Validation 1108
10.23.10 XPath Iterator 1110
10.24 Veraltete Adapter 1111
10.24.1 Atlassian Jira Connector 1112

10.24.2 Atlassian Jira Connector ReST API 1148
10.24.3 Global Parameter Initializer (früher: Custom Placeholder Initializer) (veraltet) 1150
10.24.4 MongoDB Connector 1152
10.24.5 Repository Explorer (veraltet) 1169
10.24.6 JSON Converter (veraltet) 1172
10.24.7 ZIP Compression (veraltet) 1173
10.24.8 ZIP Folder Compression (veraltet) 1174
10.24.9 ZIP in Memory Compression (veraltet) 1175

1 Einführung

1.1 Was ist ein ESB Project?

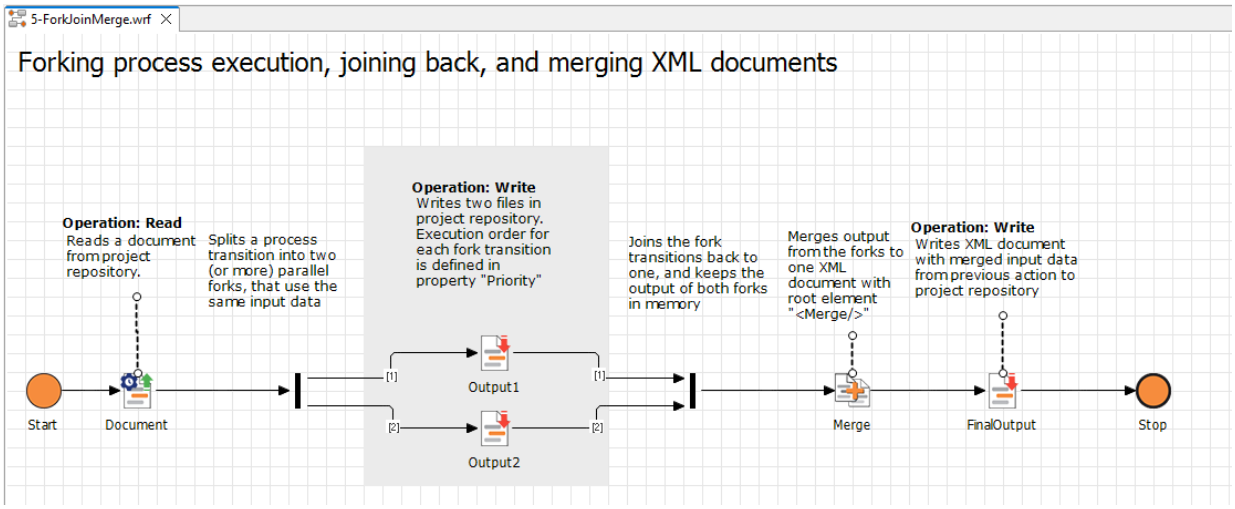
Ein ESB (Enterprise Service Bus) Project ist eine Projektart innerhalb der X4 BPMS, mit der Sie Technical Processes grafisch modellieren und ausführen können. ESB Projects enthalten eine vordefinierte Ordnerstruktur, in der beliebig viele Technical Processes (*.wrf) angelegt werden können.

1.2 Wie kann ich ESB Projects nutzen?

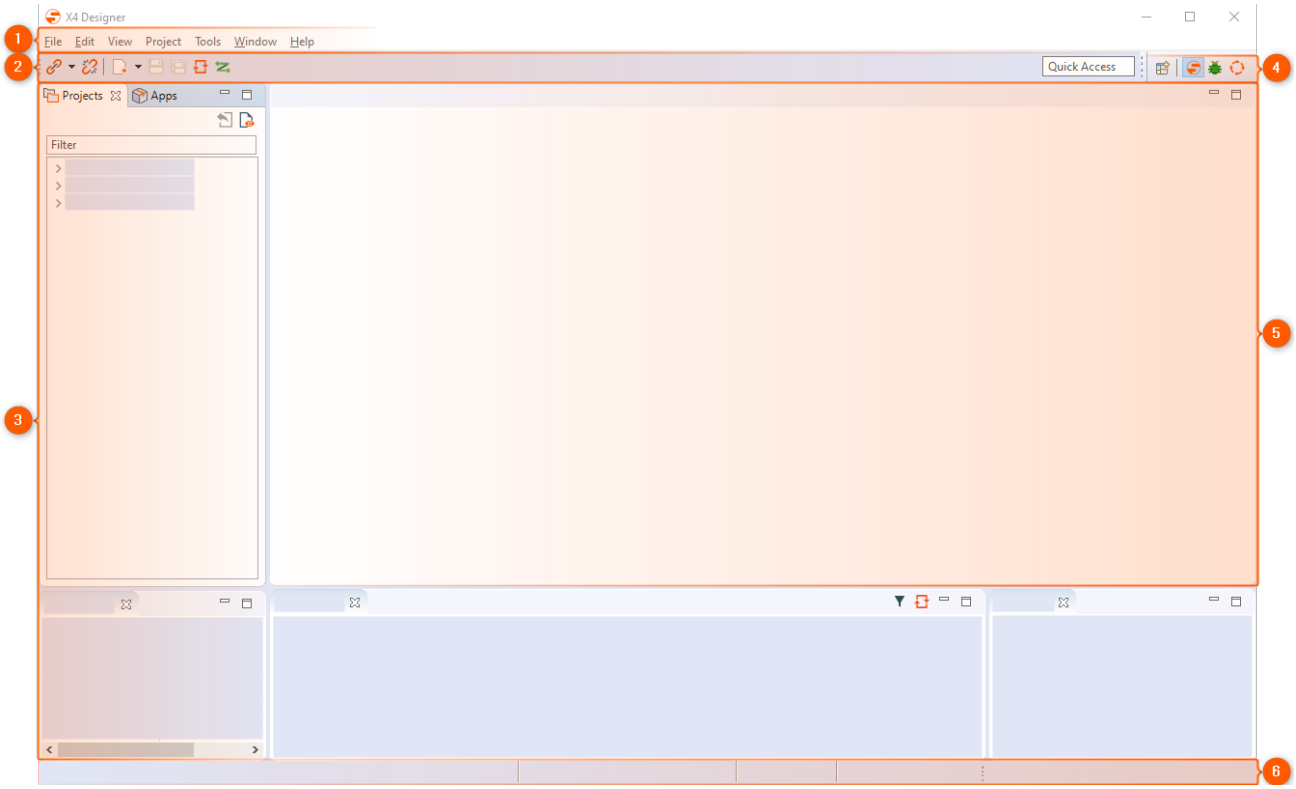
Innerhalb von ESB Projects können Sie technische Prozesse erstellen und grafisch modellieren. Alle Prozessbausteine werden dabei einfach per Drag & Drop in die Zeichnungsfläche übernommen. Dort können sie nach der Parametrierung direkt über das integrierte Debugging-Werkzeug getestet oder per Mausklick zeit- und ereignisgesteuert auf dem X4 Server ausgeführt werden.

Die Einsatzmöglichkeiten von ESB Projects sind sehr vielfältig. Beispiele für die Anwendung sind u.a. elektronischer Datenaustausch zwischen Systemen über die verfügbaren Adapter oder die Transformation von Daten über Mappings.

Forking und Merging in einem Technical Process



2 Oberfläche



1	**Menüleiste**	Über die Menüleiste können Sie verschiedene Funktionen und Editoren des X4 Designers aufrufen.
2	**Symbolleiste**	Über die Symbolleiste können Sie verschiedene Funktionen und Anzeigeoptionen aufrufen. Einige Symbole sind nur aktiv, wenn die entsprechende Funktion für das ausgewählte oder geöffnete Element verfügbar ist, siehe auch Symbolleiste.
3	**Sichten**	Sie können die angezeigten Sichten per Drag&Drop individuell anordnen.
4	**Perspektiven**	Über diese Symbole können Sie zwischen den Perspektiven X4 Designer, Debug und BPMN Designer wechseln.
5	**Editoren und Zeichnungsfläche**	In diesem Bereich werden Prozesse und Dateien zum Anzeigen und Bearbeiten in Registerkarten geöffnet.
6	**Statusleiste**	In der Statusleiste werden der verbundene Server, der angemeldete Benutzer und die Version der X4 BPMS angezeigt.

2.1 Symbolleiste

Über die Symbolleiste können Sie verschiedene Funktionen und Anzeigeoptionen aufrufen. Einige Symbole sind nur aktiv, wenn die entsprechende Funktion für das ausgewählte oder geöffnete Element verfügbar ist.

	Connect	Verbindung zum Server herstellen
	Disconnect	Verbindung zum Server trennen
	New	Neues Element hinzufügen Welche Elemente Sie über dieses Symbol hinzufügen können, hängt davon ab, in welcher Perspektive und an welcher Stelle der Baumstruktur im Repository Sie sich befinden.
	Save	Änderungen im aktuell geöffneten Fenster speichern
	Save all	Alle Änderungen speichern
	Repository Refresh	Server-seitige Aktualisierung durchführen
/	*Attach/Detach Debug*	Debugger mit der Server-Instanz verbinden/Debugger von der Server-Instanz lösen
	Start Debugging	Debugger starten
	Undo	Letzten Bearbeitungsschritt rückgängig machen
	Redo	Rückgängig gemachten Schritt wiederherstellen
	Cut	Auswahl ausschneiden
	Copy	Auswahl kopieren
	Paste	Auswahl einfügen
	Start Debugging	Optionen zur Prozessausführung aufrufen: • `Start Debugging`: Prozess debuggen (Fehler anzeigen und beseitigen) • `Start Without Debugging`: Prozess direkt ausführen • `Debug/Run Options`: Run/Debug-Einstellungen im X4 Designer öffnen

	Suspend	Debugging anhalten
	Resume	Debugging fortsetzen
	Stop Debugging	Debugging beenden
	Step Into	Prozess-Schritt ausführen (mit Anzeige der Schritte des Subprozesses)
	Step Over	Prozess-Schritt ausführen (ohne Anzeige der Schritte des Subprozesses)
	Step Return	Aus Subprozess herausspringen
	Toggle Breakpoint	Breakpoint (Debug-Haltepunkt) zu markiertem Baustein setzen oder entfernen
	Clear all Breakpoints	Alle Breakpoints in diesem Prozess löschen
	Quick Watch	Quick-Watch-Sicht öffnen, um Platzhalter, dynamische Parameter und Kontextvariablen mit ihren Werten während des Debuggens zu betrachten
	Output	Output-Sicht öffnen, um das Ergebnis jedes Prozess-Schrittes während des Debuggens zu betrachten
	Zoom Out	Ansicht verkleinern
	Zoom	Zoom-Stufe wählen
	Zoom In	Ansicht vergrößern
	Snap to grid	Raster einblenden und Bausteine am Raster ausrichten
	Show Process Quality Index	Process Quality Index einblenden

2.2 Kontextmenü

Kontextmenü für Prozessbausteine

Wenn Sie im Editor-Bereich des X4 Designers mit der rechten Maustaste auf einen Prozessbaustein klicken, stehen Ihnen je nach Bausteinart und Prozesstyp verschiedene Funktionen zur Verfügung:

Add Component	Neue Prozesskomponente hinzufügen
Layout	Ausgewählte Prozesskomponenten ausrichten: • `Align Horizontal`: Komponenten anhand des Mittelwertes horizontal ausrichten • `Align Vertical`: Komponenten anhand des Mittelwertes vertikal ausrichten • `Distribute Horizontal`: Komponenten gleichmäßig horizontal verteilen • `Distribute Vertical`: Komponenten gleichmäßig vertikal verteilen
Select All	Alle Prozesskomponenten auswählen
Copy Process Diagram to Clipboard	Prozessdiagramm zwischenspeichern
Edit Label...	Namen der Prozesskomponente bearbeiten
Undo ...	Rückgängig machen
Redo ...	Letzten Schritt wiederherstellen
Copy	Markierten Prozessbaustein kopieren
Cut	Markierten Prozessbaustein ausschneiden
Set ... document	Markierten Baustein durch ein anderes Element im Repository austauschen
Edit ...	Markierten Baustein/Datei bearbeiten (nur Modul-Bausteine und Dokumente)
Toggle Breakpoint	Debug-Haltepunkt setzen
Delete	Prozessbaustein aus dem Prozessdiagramm entfernen
Validate	Subprozess auf Gültigkeit prüfen (nur bei Subprozessen)
Preview	Prozessbaustein in separatem Editor öffnen (nur bei Modul-Bausteinen und Dokumenten)
Show input schema	Input-Schema des Subprozesses anzeigen (falls vorhanden)
Show output schema	Output-Schema des Subprozesses anzeigen (falls vorhanden)
Select in Repository Navigator	Prozessbaustein im Repository Navigator zeigen
View Call Hierarchy	Verwendungsnachweis für den markierten Prozessbaustein anzeigen
Refresh	Prozessbaustein neu aus dem Repository laden (nur bei Modul-Bausteinen und Dokumenten)

Bring to Front	Symbol in den Vordergrund des Diagramms setzen
Bring Forward	Symbol im Diagramm eine Ebene nach vorne setzen
Send Backward	Symbol im Diagramm eine Ebene nach hinten setzen
Send to Back	Symbol in den Hintergrund des Diagramms setzen
Properties	Eigenschaften des Prozessbausteins in der Properties-Sicht anzeigen
Schema Declaration...	Schema-Deklaration für den Prozess hinterlegen

Kontextmenü in der Projects-Sicht

Das Kontextmenü können Sie per Rechtsklick auf ein Element in der **Projects**-Sicht aufrufen.

New	Neue Elemente (Prozesse, Ordner, Adapter etc.) im Repository anlegen
Open / *Open With*	Markiertes Element (Prozess, Ordner, Adapter etc.) mit dem Standard-Editor oder einem bestimmten Editor öffnen
Save As Template	Markiertes Element als Vorlage speichern
Copy Full Path to Clipboard / *Copy XStore URL to Clipboard*	Pfad zur markierten Datei in die Zwischenablage kopieren (als Ordnerpfad oder als `xstore://`-URL)
Cut, Copy und Paste	Markiertes Elemente ausschneiden, kopieren oder einfügen
Delete	Markiertes Element aus dem Repository löschen (der Löschvorgang kann nicht rückgängig gemacht werden)
Rename/Refactor	Element (und falls gewünscht dessen Referenzen) umbenennen
Exclude from Project	Ressource aus dem Projekt ausschließen • Die Datei wird im Dateisystem beibehalten, ist jedoch nicht mehr Bestandteil des Projektes und wird ausgegraut dargestellt. • Die üblichen Operationen wie Öffnen, Bearbeiten, Kopieren, Löschen usw. können für ausgeschlossene Dateien nicht ausgeführt werden.
Include to Project	Ausgeschlossene Ressource wieder in das Projekt aufnehmen
Repair References	Pfade von Repository-Ressourcen in Prozessdiagrammen reparieren
Validate	Prozess auf Gültigkeit prüfen
View Call Hierarchy	Anzeigen, in welchen Prozessen dieses Element verwendet wird
Start Without Debugging	Prozess direkt ausführen
Import	Ressource ins Repository importieren
Export	Projekt oder Ressource exportieren

Clean-Up Project	Nicht verwendete Dateien auswählen und aus dem Repository löschen

2.3 Perspektiven

Eine Perspektive ist eine festgelegte Anordnung von Sichten und dem Editorbereich. Sie können per Drag&Drop individuell anordnen.

X4 Designer	Die X4 Designer-Perspektive ist die Standardansicht.
Debug	Die Debug-Perspektive ist eine optimierte Ansicht für das Debugging von Geschäftsprozessen und technischen Prozessen.
BPMN Designer	Die BPMN Designer-Perspektive ist eine optimierte Ansicht für die Modellierung von Geschäftsprozessen.

2.3.1 Perspektive wechseln

Im X4 Designer können Sie zwischen Perspektiven wechseln. Die Anordnung der Sichten bleibt dabei erhalten.

- Um die Perspektive zu wechseln, klicken Sie im oberen rechten Bildschirmbereich auf das Symbol der gewünschten Perspektive.

2.3.2 Perspektive zurücksetzen

Um eine Perspektive zurückzusetzen, gehen Sie wie folgt vor:

1. Wechseln Sie in die Perspektive.
2. Wählen Sie in der Menüleiste **Window > Reset Window Layout**.

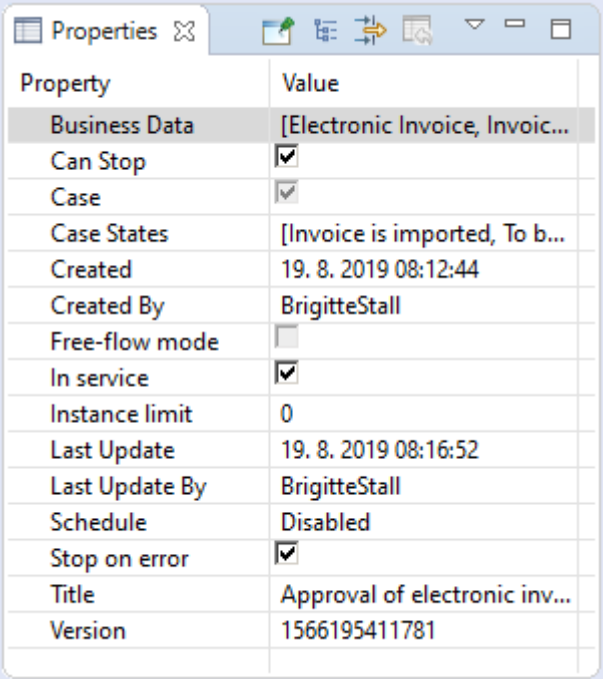
2.4 Sichten

Projects	In der Sicht **Projects** wird eine Baumstruktur des verbundenen Repositorys mit verschiedenen Elementen wie Projekten, Ordnern und Dateien angezeigt. Sie können das Kontextmenü aufrufen, indem Sie mit der rechten Maustaste auf ein Element klicken. Abhängig davon, an welcher Stelle in der Baumstruktur Sie sich befinden, ändern sich die Optionen im Kontextmenü. Über das Textfeld **Filter** über dem Projektbaum können Sie nach Dateien innerhalb des verbundenen Repositorys suchen. Platzhalter wie * oder ? sind nicht möglich. Über die Symbolleiste der Sicht **Projects** können Sie die folgenden Aktionen ausführen: • *Jump from active Editor to Repository Navigator*: Geöffnete Datei im Editor im Repository Navigator markieren • *Show or hide all files*: Inhalt des Dateisystems einblenden und ausblenden. Aus dem Projekt ausgeschlossene Dateien werden dabei ausgegraut dargestellt.
Apps	In der Sicht **Apps** werden alle X4 Apps angezeigt, die zum Bündeln von Projekten verwendet werden.

Properties

In der Sicht **Properties** können Sie die Eigenschaften von markierten Prozessbausteinen und geöffneten Prozessen anzeigen und bearbeiten.

- In der linken Spalte befinden sich die Eigenschaften, in der rechten Spalte die zugehörigen Werte.
- Für jeden Prozess kann ein Titel sowie eine Beschreibung hinterlegt werden. Diese Angaben werden in der erzeugten Dokumentation angezeigt.
- Um die Eigenschaften eines geöffneten Prozesses zu bearbeiten, klicken Sie auf einen leeren Bereich auf der Zeichnungsfläche.



Standardmäßig werden die gebräuchlichsten Eigenschaften angezeigt. Über die Symbolleiste der Sicht **Properties** können Sie jedoch zusätzliche Einstellungen vornehmen:

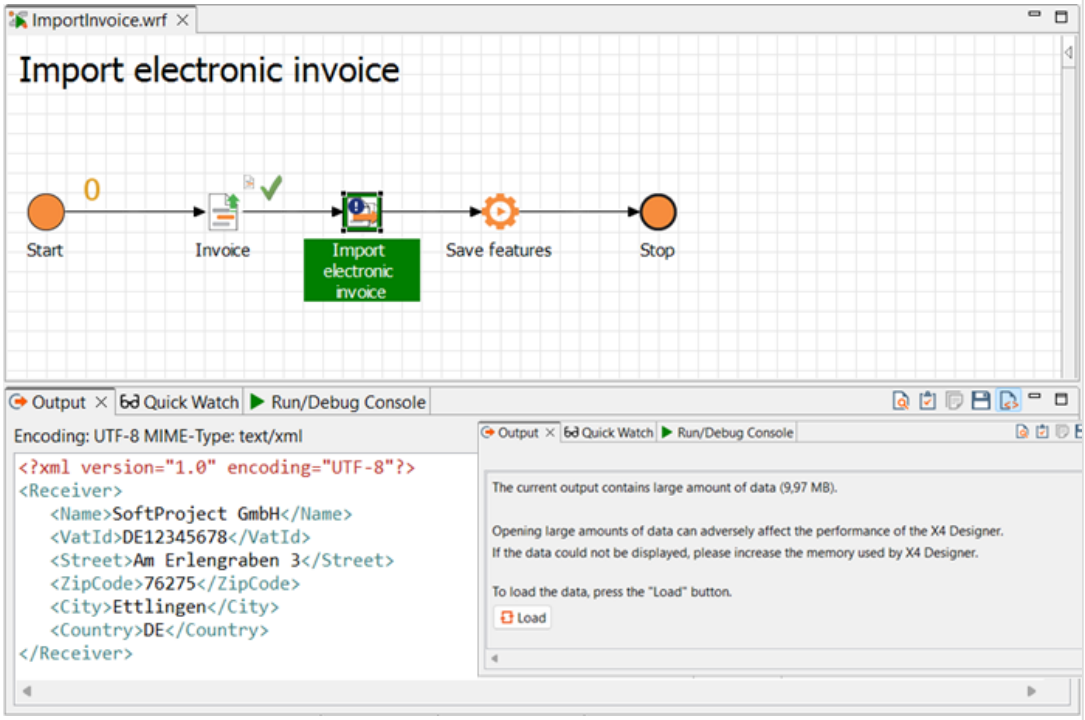
- *Pin to Selection*: Angezeigte Eigenschaften in der Sicht **Properties** anheften
- *Show Categories*: Eigenschaften nach Kategorien geordnet anzeigen
- *Show Advanced Properties*: Erweiterte Eigenschaften anzeigen
- *Restore Default Value*: Auf Standardwerte zurücksetzen
- *View Menu*: Zusatzoptionen öffnen

▶ **Run/Debug Console**	Wird ein Prozess auf Fehler überprüft bzw. simuliert, werden in der Sicht **Run/Debug Console** zu jedem ausgeführten Prozess-Schritt Debug-Informationen angezeigt. Fehlermeldungen werden rot hervorgehoben, erfolgreiche Statusmeldungen grün. Wenn ein Prozess-Schritt ein Zwischenergebnis ausgibt, wird dieses während des Debuggens in einer temporären Datei gespeichert, die Sie über den entsprechenden Verweis ansehen können (z. B. `file://1376054979039/`). Sie können die Sicht **Run/Debug Console** über das Menü **View** > **Run/Debug Console** aufrufen.

Output

In der Sicht **Output** können Sie die Ergebnisse jedes einzelnen Debug-Schritts einsehen.

Sie können die Sicht **Output** während des Debuggens über das Menü **Debug** > **Output** oder über das Symbol **Output** in der Symbolleiste aufrufen.



Wenn die ausgegebenen Daten größer als 1 MB sind, wird eine Warnung angezeigt. Sie können die Daten über eine Schaltfläche herunterladen. Große Datenmengen können die Performance des X4 Designers beeinträchtigen.

Über die Symbolleiste können Sie außerdem die folgenden Aktionen ausführen:

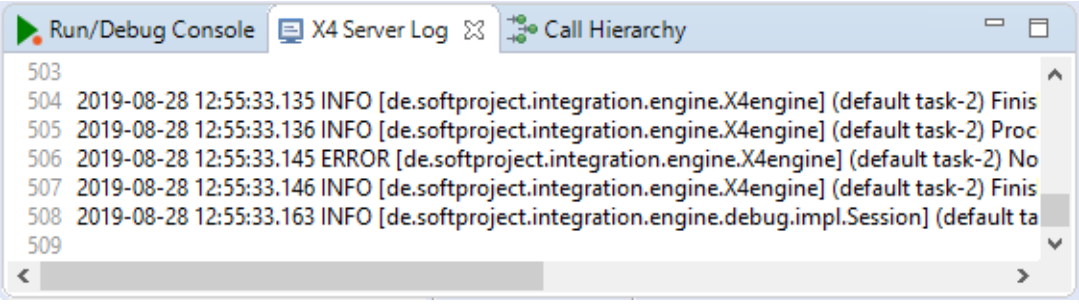
- *Find content:* Daten in der Sicht **Output** finden
- *Copy all content*: Alle Daten in der Sicht **Output** in die Zwischenablage kopieren
- *Copy selected content*: Ausgewählte Daten aus der Sicht **Output** in die Zwischenablage kopieren
- *Save content*: Daten aus der Sicht **Output** speichern
- *Format content*: Daten aus der Sicht **Output** formatieren

Hinweis

Binäre Daten werden in der Sicht **Output** als hexadezimale Daten ausgegeben. Bilder und textbasierte Inhalte werden als Bild bzw. Text ausgegeben. Durch die Ausgabe binärer Daten als hexadezimale Daten wird die Performance des X4 Designers nicht beeinträchtigt.

X4 Server Log

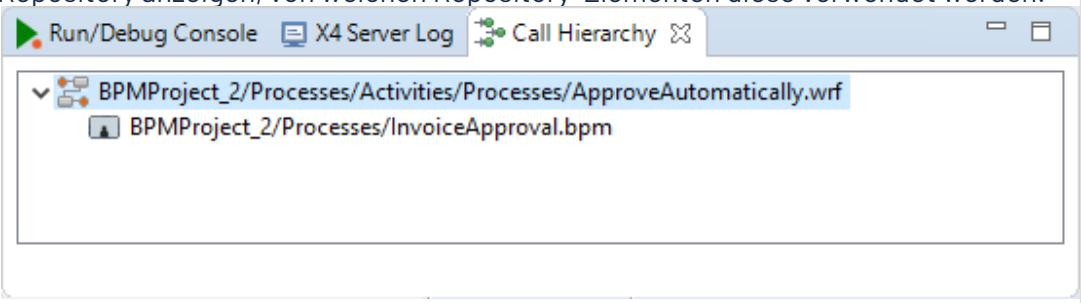
In der Sicht **X4 Server Log** können Sie den Server-Status und insbesondere Fehler im X4-Server-Log aus dem X4 Designer heraus verfolgen.



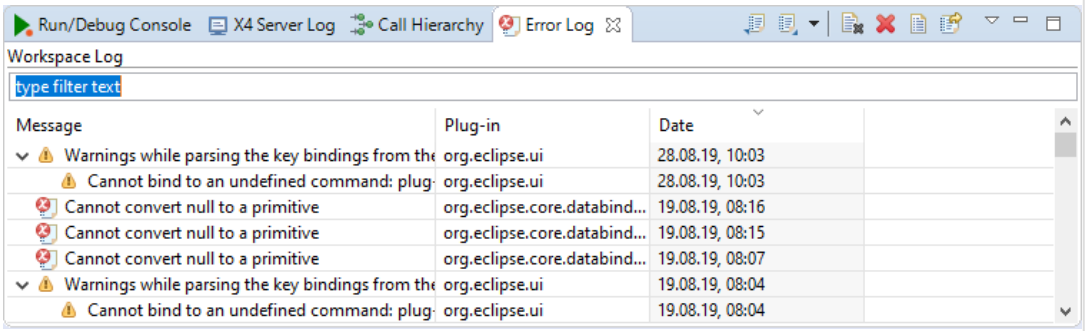
Sie können die Sicht **X4 Server Log** über das Menü **View > X4 Server Log** aufrufen.

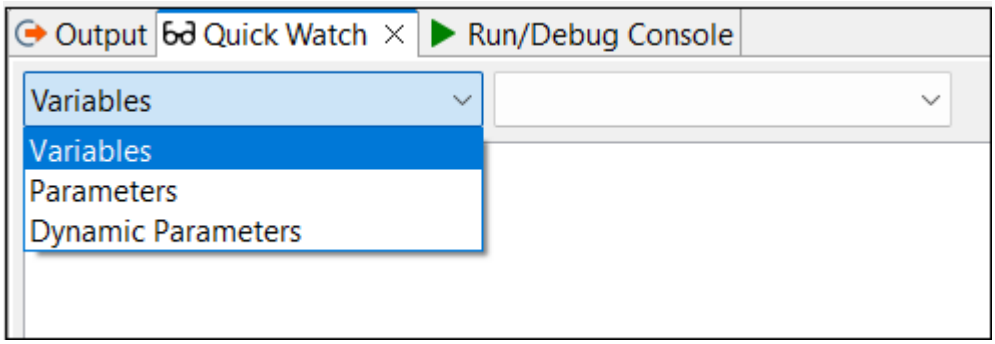
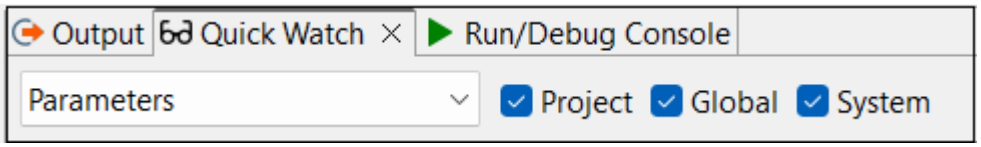
Call Hierarchy

In der Sicht **Call Hierarchy** können Sie für jeden Prozess und Prozessbaustein im Repository anzeigen, von welchen Repository-Elementen diese verwendet werden.



Wenn Sie mit der rechten Maustaste auf ein Element im Repository Navigator klicken, wird ein Kontextmenü aufgerufen. Mit **View Call Hierarchy** wird anschließend eine Auflistung aller Repository-Elemente angezeigt, die mindestens eine Referenz auf das markierte Element enthalten.

Error Log	In der Sicht **Error Log** können Sie im X4 Designer aufgetretene Fehler einsehen.  Über die Symbolleiste können Sie außerdem die folgenden Aktionen ausführen: • *Export Log*: Error Log exportieren und als `.log`-Datei speichern • *Import Log*: Error Log (`.log`) importieren • *Clear Log Viewer*: Error Log bereinigen • *Delete Log*: Error Log unwiderruflich löschen • *Open Log*: Log-Datei (`.log`) im X4 Designer öffnen • *Restore Log*: Error Log wiederherstellen • *View Menu*: Zusatzoptionen öffnen Sie können die Sicht **Error Log** über das Menü **View > Error Log** aufrufen.
Problems	In der Sicht **Problems** werden alle Fehler und Warnungen aufgelistet. Sie können die Liste der Probleme über das Filtersymbol anpassen.
Debug Sessions	In der Sicht **Debug Sessions** werden alle aktiven Debug-Sessions aufgelistet. Indem Sie in der Sicht **Debug Sessions** auf einen Prozess doppelklicken, wechseln Sie in die zugehörige Debug-Session. Aktive Debug-Sessions sind grau markiert. Mit dem Server verbundene, neue Debug-Sessions sind orange markiert.
Breakpoints	In der Sicht **Breakpoints** werden alle gesetzten Breakspoints aufgelistet. Sie können Breakpoints aktivieren, deaktivieren und löschen. Außerdem können Sie alle Breakpoints überspringen, ohne die Breakpoints zu löschen.

Quick Watch	In der Sicht **Quick Watch** können Sie Variablen, Parameter und dynamische Parameter mit ihren Werten während des Debuggens getrennt voneinander betrachten. Dazu wählen Sie in der Dropdown-Liste den entsprechenden Eintrag aus:  Ausgewählte Variablen werden nach jedem Debug-Schritt aktualisiert. Wenn Sie den Eintrag **Parameters** auswählen, können Sie über die Kontrollkästchen **Project, Global** und **System** festlegen, welche Parameter Sie anzeigen möchten:  ⚠ Enthält das Ergebnis Inhalte, die größer als 1 MB sind, wird eine Warnung angezeigt, und Sie können das Ergebnis über eine Schaltfläche herunterladen. Große Datenmengen können sich negativ auf die Leistungsfähigkeit des X4 Designers auswirken. ℹ Sie können die Sicht **Quick Watch** während des Debuggens über das Menü **Debug > Quick Watch** oder über das Quick-Watch-Symbol in der Symbolleiste aufrufen.
Outline	In der Sicht **Outline** wird das vollständige Prozessdiagramm als Abbildung angezeigt. Sie können auf einzelne Bereiche in der Abbildung klicken und so mühelos durch den Prozess navigieren, selbst wenn die Abbildung größer ist als das Designer-Fenster. ℹ Sie können die Sicht **Outline** über das Menü **View > Outline** aufrufen.
Base64	Zeichenketten Base64-kodieren oder dekodieren
Digest	Hash-Wert für Zeichenketten erzeugen
Regex	Regulären Ausdruck (*Regex*) auf eine Zeichenkette anwenden, um den Ausdruck zu testen oder um die Zeichenkette gemäß einem Muster direkt zu ersetzen

URLCodec	Zeichenketten URL-kodieren oder dekodieren

2.4.1 Sichten öffnen

> In der Perspektive **BPMN Designer** können keine weiteren Sichten geöffnet werden, da alle relevanten Sichten standardmäßig geöffnet sind.

- Wählen Sie in der Menüleiste im Menü **View** die gewünschte Sicht.

2.5 Editoren

Im X4 Designer werden Prozesse und Dateien zum Anzeigen und Bearbeiten in Registerkarten geöffnet. Der verwendete Editor ist abhängig vom geöffneten Dateityp.

Editor	Verwendung	Beschreibung
BPMN Editor	.bpm .pld	Mit dem BPMN Editor können Sie Geschäftsprozesse gemäß dem Notationsstandard BPMN 2.0 modellieren.
Datenbank-Editor	.stm	Mit dem Datenbank-Editor können Sie SQL-Abfragen für Bausteine einer Datenbankverbindung definieren, um Daten aus einer relationalen Datenbank abzufragen.
JSON Editor	.json	Mit dem JSON Editor können Sie JSON-Dokumente direkt im X4 Designer bearbeiten.
Mapping Editor	.xsl .rep	Mit dem Mapping Editor können Sie XSL-Mappings in einer grafischen Ansicht erstellen und bearbeiten.
Parameter-Editor	.param	Mit dem Parameter-Editor können Sie Projekt-Parameter und Projekt-Parameter-Gruppen erstellen, um Prozessbausteine zu parametrieren.
Process Editor	.wrf	Mit dem Process Editor können Sie technische Prozesse grafisch modellieren.
XML Editor	.xml .xhtml	Mit dem XML Editor können Sie wohlgeformte XML-Dokumente erstellen und bearbeiten.

XSD Editor	.xsd	Mit dem XSD-Editor können Sie eine XML-Schemadefinition für Ihre XML-Dokumente erstellen.

2.5.1 BPMN Editor

Mit dem BPMN Editor können Sie Geschäftsprozesse gemäß dem Notationsstandard BPMN 2.0 modellieren. Der Editor entspricht in Aufbau und Bedienweise in weiten Teilen dem Process Editor im X4 Designer, unterscheidet sich jedoch im Hinblick auf die Prozessbausteine und Symbole im Prozessdiagramm. Mit dem BPMN Editor werden die modellierten Geschäftsprozesse direkt mit den technischen Prozessen verknüpft. Das heißt: Sie können Ihre fachlichen und technischen Prozesse modellieren und über den X4 Server testen und ausführen, ohne die Entwicklungsumgebung zu wechseln.
Da auch nicht-automatisierbare Aktionen – etwa Arbeitsabläufe von Menschen – in Geschäftsprozesse eingebunden werden können, lassen sich Geschäftsprozesse nicht notwendigerweise auf dem X4 Server ausführen.

Im *Free-flow-Modus* besteht zusätzlich die Möglichkeit, den BPMN Editor als reines BPMN-Zeichenwerkzeug mit einer erweiterte BPMN-Palette zu verwenden, bei der die Funktion **Run/Debug** deaktiviert ist.

2.5.2 Datenbank-Editor

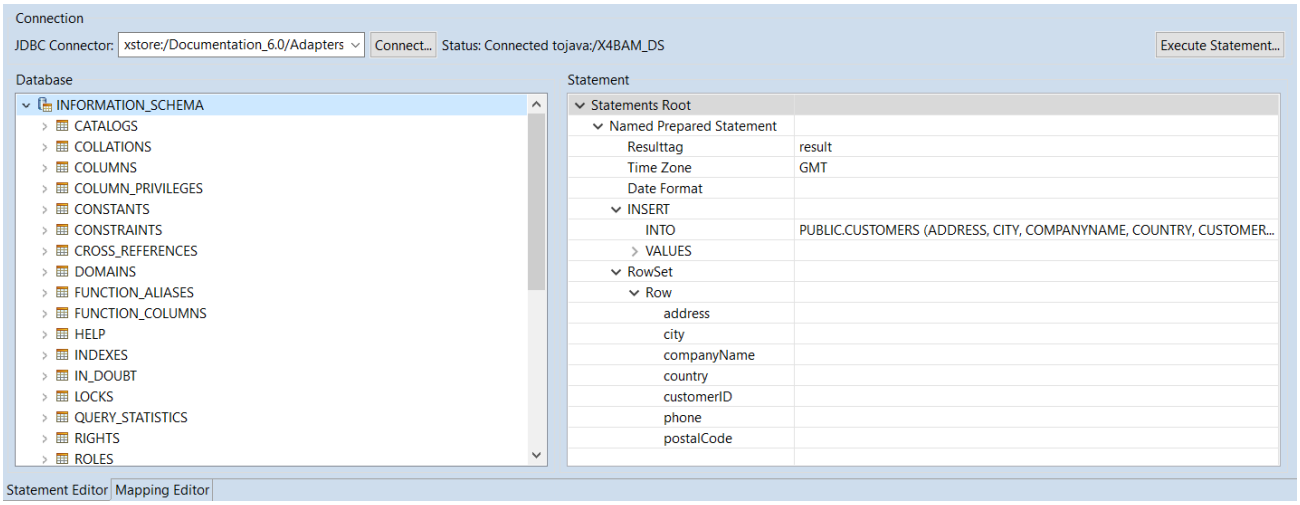
Mit dem Datenbank-Editor können Sie SQL-Abfragen für Bausteine einer Datenbankverbindung definieren, um Daten aus einer relationalen Datenbank abzufragen. Durch die logische Trennung der Prozessbausteine *SQL-Anweisung* und *Datenbankverbindung* lassen sich diese Prozessbausteine im X4 Designer flexibel einsetzen und mehrfach verwenden.

Um den Datenbank-Editor aufzurufen, legen Sie eine neue SQL-Statement-Datei (`.stm`) an, oder öffnen Sie eine bereits vorhandene SQL-Statement-Datei.

Aufbau des Datenbank-Editors

Der Datenbank-Editor besteht aus den folgenden Bereichen:

- **Statement Editor**: In diesem Bereich können Sie eine Verbindung zu einer Datenbank aufbauen und die Struktur der verbundenen Datenbank anzeigen. Mit dieser Struktur können anschließend SQL-Statements für die Datenbank erstellt werden.
- **Mapping Editor**: In diesem Bereich können Sie für das im Statement Editor zuvor erstellte Statement ein Mapping von Elementen eines Input-XML-Dokuments auf die Named Parameter des Statements definieren. Aus diesem Mapping können Sie anschließend ein XSL-Mapping erstellen, das mit einem passenden Input-XML-Dokument eine dynamisch erzeugte Eingabe für den JDBC Connector erzeugt.



2.5.2.1 Statement Editor

Der Statement Editor ist in folgende Bereiche unterteilt:

Connection-Bereich	In diesem Bereich können Sie über einen *JDBC Connector* eine Verbindung zu einer Datenbank aufbauen. • **JDBC Connector**: xstore-URL zu einem *JDBC Connector* ⓘ • Um eine Verbindung aufbauen zu können, muss der Adapter innerhalb des Parameters `JndiName` einen gültigen JNDI-Namen enthalten. • In der Dropdown-Liste werden die letzten acht verbundenen URLs zur Auswahl angeboten. • **Connect**: Mit der Datenbank verbinden ⓘ Unter **Status** wird der Verbindungstatus angezeigt. • **Execute Statement**: Statement ausführen
Database-Bereich	In diesem Bereich wird nach dem Verbindungsaufbau die Struktur der Datenbank angezeigt. Für die Darstellung der Struktur werden die folgenden Symbole verwendet: • Datenbankschema • Datenbanktabelle • Primärschlüssel • Primärschlüssel/Fremdschlüssel • Fremdschlüssel

Statement-Bereich

In diesem Bereich können Sie über das Kontextmenü verschiedene Statements für die verbundene Datenbank anlegen:

- **New Named Prepared Statement**: Im `Statements Root` ein neues `Named Prepared Statement` anlegen
- **Edit Named Prepared Statement**: `Named Prepared Statement` bearbeiten
- **New RowSet**: Neues `RowSet` anlegen.
  Innerhalb eines `Named Prepared Statement` kann lediglich ein `RowSet` existieren. Dieses wird beim Anlegen des `Named Prepared Statement` automatisch mit angelegt. Ein neues `RowSet` kann nur angelegt werden, wenn das bestehende `RowSet` gelöscht wurde.
- **New Row**: Neue Zeile innerhalb eines `RowSet` hinzufügen
- **New Named Parameter Value**: Einen neuen Named Parameter-Wert innerhalb einer Zeile hinzufügen
- **Edit Named Parameter**: Named Parameter bearbeiten
- **New SELECT Statement**: Ein neues `SELECT`-Statement anlegen
- **Edit SELECT**: `SELECT`-Befehl bearbeiten
- **Edit FROM**: `FROM`-Befehl innerhalb eines `SELECT`-Statements bearbeiten
- **New INSERT Statement**: Ein neues `INSERT`-Statement anlegen
- **Edit INTO**: `INTO`-Befehl bearbeiten
- **New VALUE Entry**: Neuen Eintrag unter `VALUE` anlegen
- **Edit VALUE**: Wert eines `VALUE`-Eintrags bearbeiten
- **New UPDATE Statement**: Ein neues `UPDATE`-Statement anlegen
- **Edit UPDATE**: `UPDATE`-Befehl bearbeiten
- **New SET Entry**: Neuen Eintrag unter `SET` anlegen
- **Edit SET Entry**: Eintrag unter `SET` bearbeiten
- **New DELETE Statement**: Ein neues `DELETE`-Statement anlegen
- **Edit FROM**: `FROM`-Befehl bearbeiten
- **Edit WHERE**: `WHERE`-Bedingung bearbeiten
- **New Column**: Neue Spalte innerhalb eines `INSERT`- oder `SELECT`-Befehls einfügen
- **New Table**: Neue Tabelle innerhalb eines `UPDATE`- oder `SELECT`-Befehls einfügen
- **Delete**: Element löschen

> Weitere Informationen zum Anlegen und Bearbeiten von Statements finden Sie im Abschnitt Statement für eine verbundene Datenbank erstellen.

Drag&Drop-Aktionen

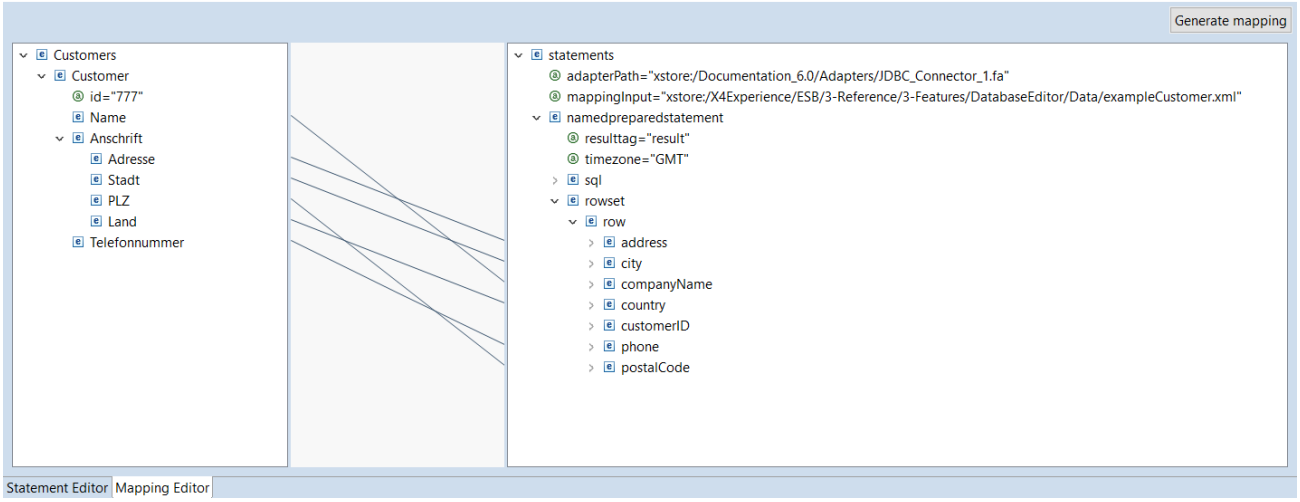
Im Statement-Bereich können Sie die folgenden Drag&Drop-Aktionen ausführen:

SQL-Statement	Datenbank-Element	Statement-Element	Aktion
SELECT	Column	SELECT	Die auszuwählende Spalte hinzufügen Steht die Tabelle der Spalte noch nicht im FROM, wird diese hinzugefügt.
	Table	FROM	Tabelle hinzufügen, aus der ausgewählt werden soll
UPDATE	Column	SET Entry	Spalte als Eintrag setzen Steht die Tabelle der entsprechenden Spalte noch nicht im UPDATE, wird diese hinzugefügt.
	Table	UPDATE	Die zu aktualisierende Tabelle setzen
INSERT	Table	INTO	Tabelle setzen, in die geschrieben werden soll Ist bereits eine Tabelle vorhanden, wird diese ersetzt. Dabei werden auch bereits gesetzte Spalten (Columns) gelöscht.
	Column	INTO	Spalte setzen, in die geschrieben werden soll Dies ist nur möglich, wenn die Spalte aus der gesetzten Tabelle stammt.
DELETE	Table	FROM	Die zu löschende Tabelle setzen Ist bereits eine Tabelle vorhanden, wird diese ersetzt.

2.5.2.2 Mapping Editor

Mit dem Mapping Editor können Sie für die zuvor im Statement Editor angelegten Statements per Drag&Dropp Mappings von einem Input-XML-Dokument auf die Named Parameters der Statements konfigurieren und anschließend erstellen. Diese Mappings erzeugen mit einem passenden Input-XML-Dokument eine dynamisch erzeugte Eingabe für den JDBC Connector.

Der Mapping Editor ist in folgende Bereiche unterteilt:



Quelldokument-Bereich	Auf der linken Seite befindet sich in der Quelldokument-Bereich. In diesem Bereich können Sie die Input-XML-Datei per Drag&Drop ablegen und anzeigen. Über das Kontextmenü können Sie den gewählten Knoten wie folgt bearbeiten: • **Collapse**: Knoten mit aufgeklappten Unterelementen zuklappen • **Expand**: Unterelemente des Knotens aufklappen • **Remove Input**: Input-XML-Dokument entfernen
Verknüpfungsbereich	In der mittleren Spalte befindet sich der Verknüpfungsbereich. In diesem Bereich werden die `xsl:value-of`-Verknüpfungen zwischen dem Element aus dem Quelldokument und dem Named Parameter im Statement als Linien visualisiert.
Statements-Bereich	Auf der rechten Seite befindet sich der Statements-Bereich. In diesem Bereich wird die gesamte Struktur der Statement-Datei angezeigt. Dabei können Mappings vom Input-XML auf die Kind-Elemente eines Row-Elements erstellt werden.

2.5.3 JSON-Editor

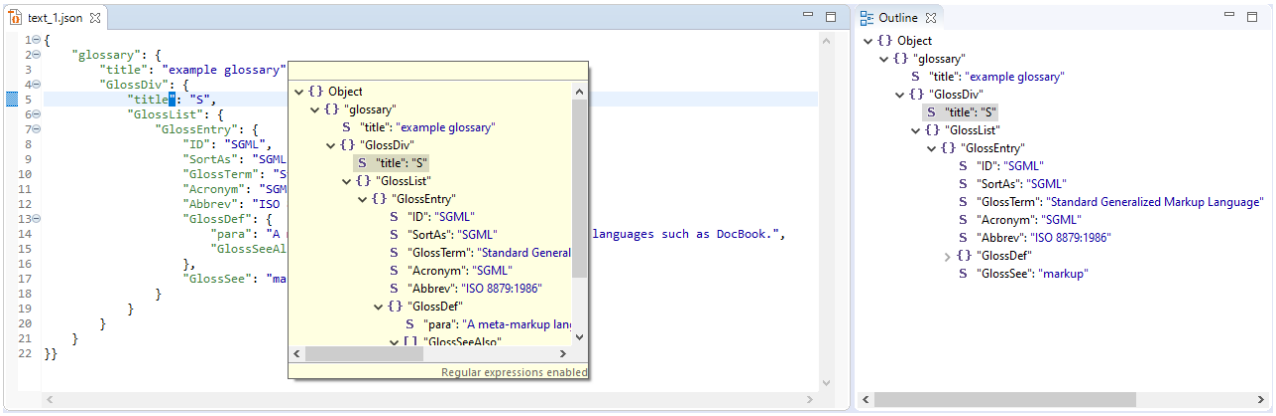
Mit dem JSON Editor können Sie JSON-Dokumente direkt im X4 Designer bearbeiten.

> ⓘ Um den JSON Editor zu starten, legen Sie entweder ein neues JSON-Dokument an oder öffnen ein bereits vorhandenes Dokument.

In der Sicht **Outline** wird die Struktur des geöffneten JSON-Dokuments angezeigt. Zudem können Sie über das Kontextmenü **Quick Outline** oder über die Tastenkombination `STRG+O` die Sicht **Quick Outline** öffnen, um innerhalb des JSON-Dokuments nach Schlüsseln und Werten zu suchen.

Tastenkombinationen

`STRG+O`	Sicht **Quick Outline** öffnen
`STRG+Umschalttaste+F`	JSON-Code formatieren



Die Formatierung des JSON-Codes sowie die Farben zur Syntaxhervorhebung können Sie unter **Tools > Options > JSON Editor** ändern. Siehe auch JSON-Editor konfigurieren.

2.5.4 Mapping Editor

Mit dem Mapping Editor können Sie XSL-Mappings in einer grafischen Ansicht erstellen und bearbeiten. Eine XML-Datenstruktur kann hier mithilfe von XSLT in eine andere XML-Datenstruktur oder in ein XSL-FO-Dokument transformiert werden. Der Mapping Editor unterstützt Sie beim Erstellen eines XSL-Mappings durch eine grafische Oberfläche und durch eine automatische Gültigkeitsprüfung des Stylesheets sowie der XPath-Ausdrücke.

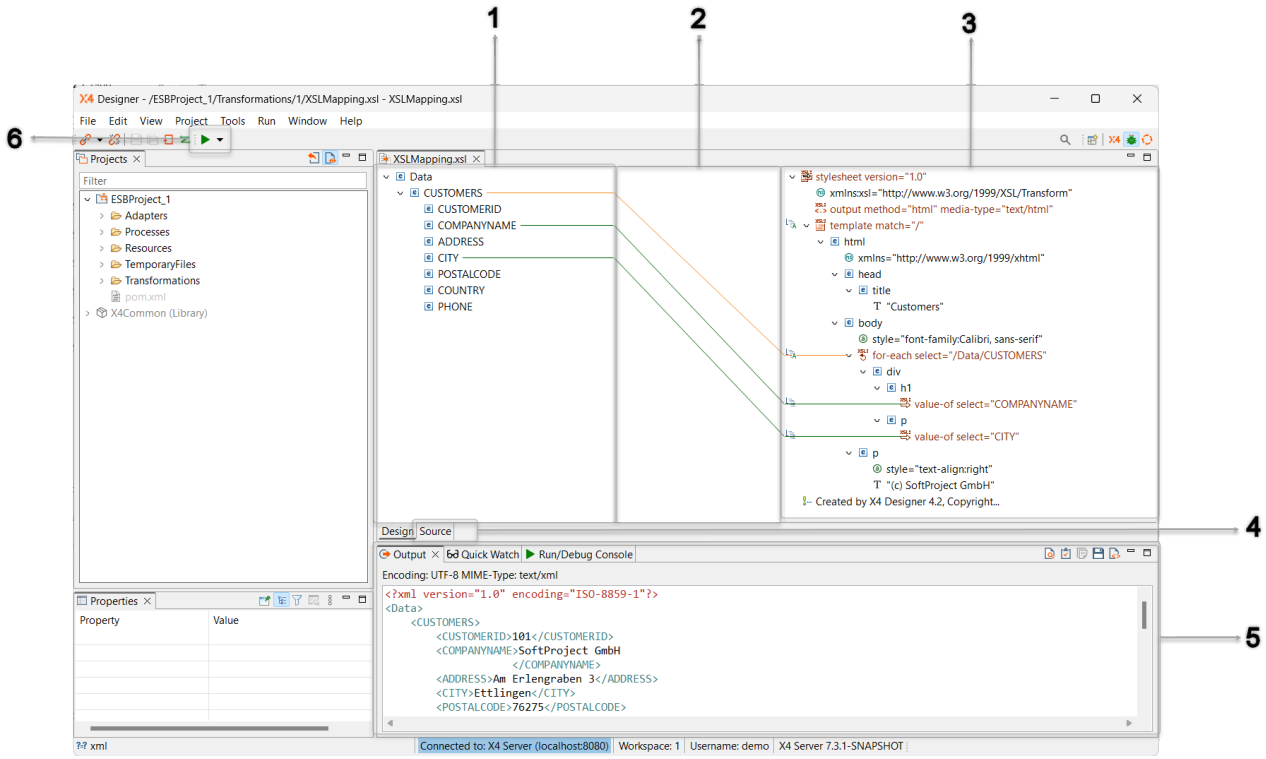
Bei einem XSL-Mapping sind drei Dokumente involviert:

- XML-Quelldokument (oder ein XML-Schema mit der Strukturinformation)
- XSLT-Stylesheet (dieses wird im Mapping Editor erstellt)
- XML-Ausgabedokument

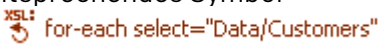
Um den Mapping Editor zu starten, legen Sie entweder ein neues Mapping an oder öffnen ein bereits vorhandenes XSLT-Mapping.

Aufbau des Mapping Editors

Der Mapping Editor ist in folgende Bereiche unterteilt:



1	**Quelldokument-Bereich**	In diesem Bereich wird das zu transformierende Quelldokument oder eine XML-Schema-Datei (`.xsd`) angezeigt. Über das Kontextmenü können Sie den gewählten Knoten wie folgt bearbeiten: • **Copy XPath for this element to Clipboard**: XPath-Ausdruck zum entsprechenden Element in die Zwischenablage kopieren und an anderer Stelle verwenden • **Assign Input**: Input-XML-Dokument zuweisen • **Remove Input**: Input-XML-Dokument entfernen • **Paste Input**: Input-XML-Dokument aus der Zwischenablage einfügen
2	**Verknüpfungsbereich**	In diesem Bereich werden die Verknüpfungen zwischen Knoten im Quelldokument und XSL-Stylesheet als verschiedenfarbige Linien visualisiert. ⓘ Unter Tastenkombinationen im Mapping Editor finden Sie nützliche Tastenkombinationen für die Arbeit mit XSL-Stylesheets in der grafischen Ansicht des X4 Mapping Editors.

3	**Stylesheet-Bereich**	In diesem Bereich bearbeiten Sie den Hauptbestandteil eines Mappings, das XSL-Stylesheet. Das Stylesheet wird in einer Baumstruktur angezeigt und bietet über das Kontextmenü Zusatzfunktionen für die XSL-Bearbeitung.
		
		Die verschiedenen Bestandteile des XSL-Stylesheets werden in der Baumansicht mit Symbolen und farblichen Hervorhebungen dargestellt:
		
		• Knoten mit XSL-Anweisungen sind durch eine braune Schriftfarbe und ein entsprechendes Symbol gekennzeichnet, z. B.  .
		• Knoten mit einem XPath-Ausdruck sind durch ein entsprechendes XPath-Symbol ( ) gekennzeichnet. Um den XPath-Ausdruck zu bearbeiten, klicken Sie auf dieses Symbol.
		• Virtuelle Knoten (aus der Quellstruktur übernommene, aber noch nicht im Stylesheet explizit vorhandene Knoten) werden grau dargestellt.
		
		Über das Kontextmenü können Sie im Stylesheet-Bereich den gewählten Knoten wie folgt bearbeiten:
		
		• **Edit XSL Element**: XSL-Element bearbeiten
		• **Rename**: Knoten umbenennen
		• **Element**: Element einfügen
		• **Attribute**: Attribut einfügen
		• **Edit**: (bei Processing Instructions, Namespace-Deklarationen): Knoteninhalt bearbeiten
		• **XSL Element**: XSL-Element einfügen
		  • `as Child`: als Kind-Knoten
		  • `Before`: als Knoten vor dem markierten Knoten
		  • `After`: als Knoten hinter dem markierten Knoten
		• **Add other nodes**: andere Knoten einfügen (`Namespace`, `Text`, `Comment`, `Processing Instruction`)
		• **Add here**: (nur bei virtuellen Knoten): virtuellen Knoten in tatsächlichen Knoten umwandeln
		• **Add Processing Instructions**: Processing Instructions hinzufügen, um Parameter dynamisch zu setzen
		  • `Condition State` für Statusinformationen
		  • `Log Info` für Log-Nachrichten
		  • `Dynamic Parameter` zum Setzen von dynamischen Werten
		  • `No Result` für leere Mapping-Ausgaben
		• **Add Parameter**: System-, Projekt- oder globale Parameter einfügen, siehe Parameter erstellen und verwenden
		• **Add Parent**: Eltern-Element (übergeordnetes Element) einfügen
		• **Cut/Copy/Paste**: Auswahl ausschneiden, kopieren und einfügen

		• **Delete:** Knoten löschen • **Expand/Collapse**: Unterelemente des Knotens aufklappen bzw. zuklappen • **Assign Virtual Nodes**: virtuellen Knoten aus einem XML-Dokument einfügen • **Remove Virtual Node**: eingefügten virtuellen Knoten entfernen Mit dem **XSL Element Creation Wizard** können Sie einfach und schnell XSL-Mappings erstellen. Der Wizard bietet neben der kompletten Palette an XSLT 2.0-Elementen auch die Möglichkeit, die Mappings bequem über das Kontextmenü anzulegen. Über den Wizard lässt sich festlegen, ob das neue Element vor oder nach dem ausgewählten Element oder als Kind des ausgewählten Elements angelegt werden soll. Funktionen wie Autovervollständigung oder die Anzeige aller verfügbaren und erlaubten XSL-Elemente ermöglichen schnelleres Arbeiten. Zudem können die Attribute des XSL-Elements gesetzt bzw. nach dem Anlegen bearbeitet werden. Zwingend erforderliche Attribute werden dabei automatisch übernommen, optionale Attribute können je nach Bedarf ausgewählt und befüllt werden. Bei Attributen, die mehrere Optionen zur Auswahl haben, wird eine Liste mit den verfügbaren Werten bereitgestellt.
4	**Source- und Design-Ansicht**	Der Mapping Editor verfügt über zwei Ansichten, zwischen denen Sie über zwei Registerkarten wechseln können: • **Design**: In der dreispaltigen Design-Ansicht wird das XSLT-Stylesheet als Baum dargestellt. Hier können Sie die Knoten des Quelldokuments den Knoten im XSLT-Stylesheet grafisch zuordnen. • **Source**: In dieser Quellcode-Ansicht können Sie das XSLT-Stylesheet direkt im XML-Editor bearbeiten.
5	**Output**	Das Ergebnis von lokal ausgeführten XSL-Mappings wird in der Sicht **Output** angezeigt. Außerdem können Sie über die Symbolleiste die folgenden Aktionen ausführen: • *Save content of output view*: Daten aus der Sicht **Output** speichern • *Format the content of the output view*: Daten aus der Sicht **Output** formatieren

6	**Run Transformation**	Über ▶ **Run Transformation** wird das XSL-Mapping lokal ausgeführt, und das Transformationsergebnis wird in der Sicht **Output** (5) angezeigt.

2.5.4.1 Tastenkombinationen im Mapping Editor

Hier finden Sie nützliche Tastenkombinationen für die Arbeit mit XSL-Stylesheets in der grafischen Ansicht des X4 Mapping Editors.

Aktionen per Drag&Drop

Werte mit dem Zielknoten verknüpfen mit `xsl:value-of`	Knoten im Quelldokument per Drag&Drop auf den Zielknoten ziehen
Mehrere Werte im Zielknoten verknüpfen mit `xsl:value-of`	**Alt**-Taste drücken und alle gewünschten Knoten im Quelldokument auf den Zielknoten ziehen *Ergebnis:* im `select`-Attribut des erzeugten `xsl:value-of`-Elements werden alle verknüpften Elemente nacheinander hinterlegt und können anschließend mit einer XPath-Funktion wie `concat()` oder logischen Operatoren verknüpft werden. *Beispiel:* `<xsl:value-of select="Element1 Element2">`
Knoten kopieren mit `xsl:copy-of`	**Strg**-Taste drücken und Knoten im Quelldokument auf den Zielknoten ziehen
For-Each-Schleife zu Knoten erstellen mit `xsl:for-each`	**Shift**-Taste drücken und Knoten im Quelldokument auf den Zielknoten ziehen

	Funktion	**Tastenkombination**
Basisfunktionen	Element bearbeiten	`Enter`
	XSL-Transformation ausführen	`F9`
	Element einfügen	`Einfg`
	Attribut einfügen	`Alt+Einfg`
	Text einfügen	`Strg+Einfg`
	Umschließendes Eltern-Element einfügen	`Strg+Shift+Einfg`
	Element umbenennen	`Alt+Shift+R`
	Markierten Knoten entfernen	`Entf`

	Code formatieren/einrücken (in der Source-Ansicht)	`Strg+Shift+F`
Knoten kopieren	Markierten Knoten als vorangehenden Knoten einfügen	`Strg+Alt+Nach oben`
	Markierten Knoten als nachfolgenden Knoten einfügen	`Strg+Alt+Nach unten`
	Markierten Knoten als vorangehenden Eltern-Knoten einfügen	`Strg+Alt+Nach links`
Knoten verschieben	Markierten Knoten als vorangehenden Knoten verschieben	`Alt+Nach oben`
	Markierten Knoten als nächsten Knoten verschieben	`Alt+Nach unten`
	Markierten Knoten als vorangehenden Eltern-Knoten verschieben	`Alt+Shift+Nach links`
Im Stylesheet-Baum navigieren	Zum Eltern-Element wechseln	`Strg+Shift+Nach links`
	Zum nächsten Geschwister-Knoten wechseln	`Strg+Shift+Nach unten`
	Zum vorangehenden Element wechseln	`Strg+Shift+Nach oben`
	Verbundene Knoten in Quelldokument-Bereich und Stylesheet-Bereich markieren	`Shift+Einfg`
	Letzte Markierung wiederherstellen	`Alt+Shift+Nach unten`

2.5.5 Process Editor

Mit dem Process Editor können Sie technische Prozesse grafisch modellieren. Außerdem können Sie für jeden technischen Prozess und jedem einzelnen Prozessbaustein festlegen, wie diese ausgeführt werden sollen.

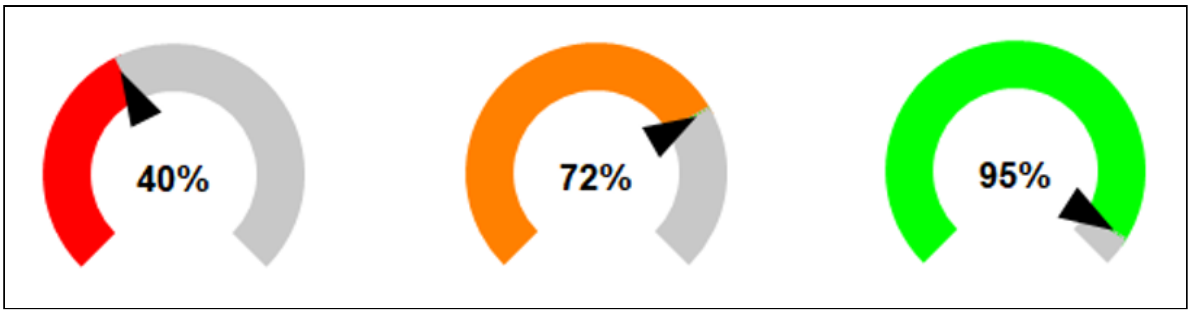
Mit dem Process Editor können Sie unter anderem:

- Verbindungen zwischen verschiedenen Backend-Systemen herstellen,
- Daten in Echtzeit transformieren,
- Geschäftsregeln im modellierten Prozess implementieren und
- modellierte Prozesse debuggen und auf dem X4 Server ausführen

2.5.5.1 Process Quality Index

ⓘ Der Process Quality Index ist ein experimentelles Feature. Die Funktionalität ist derzeit noch eingeschränkt und wird fortwährend weiterentwickelt.

Der Prozess Quality Index ist ein grafisches Hilfswerkzeug, das angibt, wie qualitativ hochwertig ein ESB-Prozess im Hinblick auf Kriterien wie die Einhaltung von Standards und die Übersichtlichkeit modelliert wurde.



Gewichtung der Kriterien

Kriterien	Bedingung	Gewichtung
Anzahl der Elemente	**100%** → **< 15** Elemente **75%** → **< 20** Elemente **50%** → **< 25** Elemente **25%** → **< 30** Elemente **0%** → **>= 30** Elemente	45%
Anzahl der Stopp-Elemente	**100%** → **< 3** Elemente **75%** → **< 4** Elemente **50%** → **< 5** Elemente **25%** → **< 6** Elemente **0%** → **>= 6** Elemente	15%
Labels der Elemente sind nicht leer	**wahr** → **100%** **falsch** → **0%**	10%
Prozesstitel wurde gesetzt (nicht Name der Prozessdatei)	**wahr** → **100%** **falsch** → **0%**	5%
Labels der Elemente wurden angepasst (nicht Dateiname)	**wahr** → **100%** **falsch** → **0%**	5%

Kriterien	Bedingung	Gewichtung
Keine Elemente auf der rechten Seite von Stopp-Elementen	**wahr → 100%** **falsch → 0%**	5%
Keine Elemente auf der linken Seite von Start-Elementen	**wahr → 100%** **falsch → 0%**	5%
Keine überlappenden Elemente	**wahr → 100%** **falsch → 0%**	5%
Flow Direction Index		5%
Maximum		**100%**

2.5.5.2 Tastenkombinationen im Process Editor

Hier finden Sie nützliche Tastenkombinationen für die Arbeit mit dem Process Editors.

	Funktion	Tastenkombination oder Aktion
Prozess ausführen und debuggen	Prozess ausführen	`F9`
	Prozess auf Fehler prüfen (und ggf. aktiven Debugging-Vorgang abbrechen)	`F4`
	Beim Debuggen Einzelschritt ausführen und anzeigen (auch in Subprozessen)	`F5`
	Beim Debuggen Prozess-Schritt ausführen (ohne Anzeige des Schritte des Subprozesses)	`F6`
	Beim Debuggen aus Subprozess herausspringen	`F7`
	Debugging anhalten	`Strg+F8`
	Debugging fortsetzen	`F8`

	Debugging abbrechen	`Strg+F2`
Prozessbausteine und Übergänge einfügen	Prozessbaustein zwischen zwei verbundenen Bausteinen aus dem Repository einfügen	Datei per Drag&Drop aus dem Repository auf den Übergang ziehen. Wenn die Linie des Übergangs dick wird, Maustaste loslassen.
	Prozessbaustein von der Übergangslinie herunterziehen	`Alt`-Taste drücken und Prozessbaustein auf der Zeichnungsfläche verschieben
	Prozessbaustein entfernen und Übergangslinie behalten	`Alt+Entf`
	Prozessbaustein durch eine Datei aus dem Repository ersetzen	Datei per Drag&Drop aus dem Repository auf den zu ersetzenden Prozessbaustein ziehen
	Prozessbaustein hinter einem unverbundenen Baustein einfügen und dabei automatisch einen Übergang herstellen	Unverbundenen Prozessbaustein markieren, `Shift`-Taste drücken und die Datei aus dem Repository hinter den markierten Prozessbaustein ziehen
Elemente bearbeiten	Alles auf der Zeichnungsfläche auswählen	`Strg+A`
	Prozessbaustein ausschneiden	`Strg+X`
	Prozessbaustein kopieren	`Strg+C`
	Prozessbaustein einfügen	`Strg+V`
	Prozessbaustein entfernen	`Entf`
	Bearbeitungsschritt rückgängig machen	`Strg+Z`
	Bearbeitungsschritt wiederherstellen	`Strg+Y`
	Prozessbaustein bearbeiten	Auf Symbol doppelklicken
	Editor schließen	`Strg+W`
	Bearbeitetes Element speichern	`Strg+S`
	Datei im Repository umbenennen	`F2`
	Datei im Repository verschieben	Datei per Drag&Drop in den Ordner ziehen
	Neu laden	`F5`
Ansicht bearbeiten	Prozessdiagramm vergrößern (zoomen)	`Strg++`
	Prozessdiagramm verkleinern	`Strg+-`

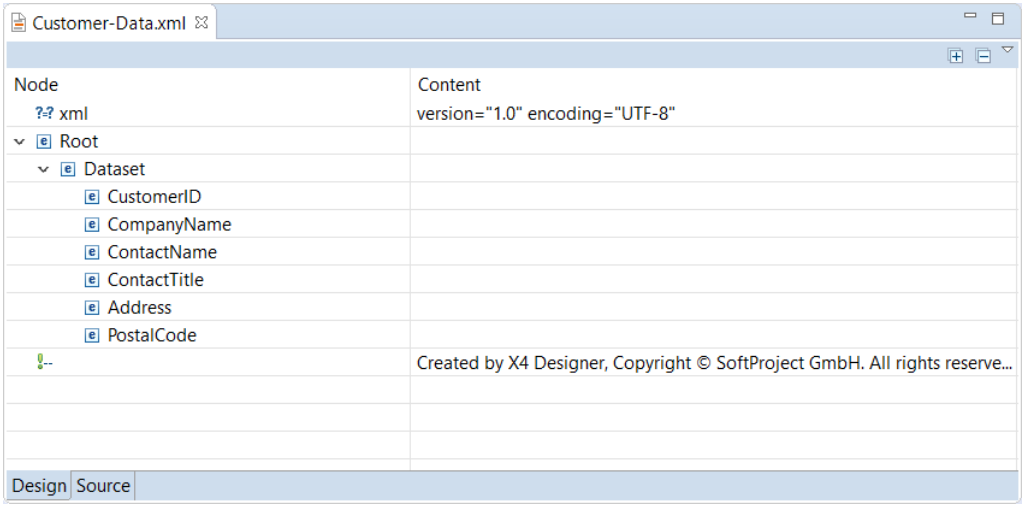
2.5.6 XML-Editor

Mit dem XML Editor können Sie wohlgeformte XML-Dokumente erstellen und bearbeiten.

> ⓘ Um den XML Editor zu starten, legen Sie entweder ein neues XML-Dokument an oder öffnen ein bereits vorhandenes Dokument.

Aufbau des XML-Editors

- Zum Bearbeiten von XML-Dokumenten stehen zwei Sichten zur Verfügung: die Sicht **Design** und die Sicht **Source**. Über die Registerkarten am unteren Rand des Editors können Sie zwischen den Sichten wechseln.

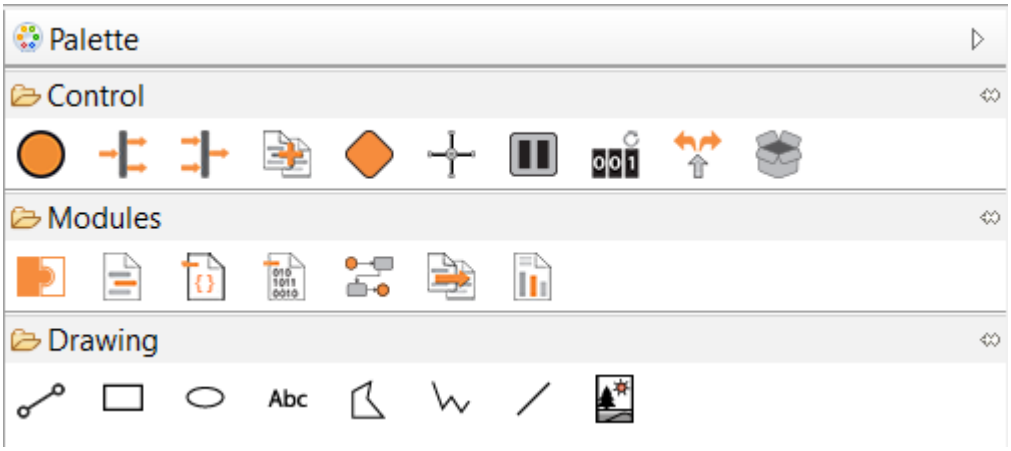


Design	In der Sicht **Design** wird die Baumstruktur des geöffneten XML-Dokuments tabellarisch dargestellt. In der linken Spalte befinden sich die verschiedenen Knoten, die Sie auf- und zuklappen können, sofern Kindknoten existieren. In der rechten Spalte werden die Inhalte der Knoten angezeigt. Über das Kontextmenü können Sie die folgenden Aktionen ausführen: • **Remove**: Markierten Knoten entfernen • **Add DTD Information**: Strukturdefinition in Form einer DTD (*Document Type Definition*) hinzufügen • **Edit Namespaces**: Namensräume bearbeiten • **Add Attribute**: Neues Attribut hinzufügen • **Add Child**: Neuen Kindknoten (Element, Processing Instruction usw.) hinzufügen • **Add Before**: Neuen Knoten vor dem markierten Knoten einfügen • **Add After**: Neuen Knoten hinter dem markierten Knoten einfügen
Source	In der Sicht **Source** können Sie den Quelltext von XML-Dokumenten bearbeiten. Dabei unterstützt Sie der Editor durch folgende Funktionen: • Syntax-Highlighting: Die verschiedenen Knotentypen werden farblich hervorgehoben. • Wohlgeformtheit: Beim Speichern erfolgt eine Prüfung auf Wohlgeformtheit des XML-Dokuments. • Eingabehilfe: Der Editor macht Vorschläge, wenn Sie neue Elemente einfügen möchten. • Pretty-Print: Über das Menü **Source > Format** können Sie Ihr XML-Dokument mit Einrückungen und Zeilenumbrüchen lesbarer formatieren.

3 Prozessbausteine und ihre Verwendung

3.1 Die Palette

Am rechten Rand der Zeichnungsfläche befindet sich eine ausklappbare Werkzeugleiste, die *Palette*. Sie enthält verschiedene Prozessbaustein-Symbole, die Sie per Drag&Drop in die Zeichnungsfläche ziehen können.



Wenn Sie den Mauszeiger über einen in der Zeichnungsfläche platzierten Baustein bewegen, dann werden oberhalb des Bausteins verschiedene Symbole angezeigt.

> ⚠ Es sind nicht alle Symbole bei allen Bausteinen verfügbar!

Symbol	Funktion/Bedeutung
+	Neuen Baustein nach dem aktuellen Baustein hinzufügen
	Typ des Bausteins von `Resource` in `Buffer` ändern
	Typ des Bausteins von `Buffer` in `Resource` ändern ⚠ Für Dateien, die im Ordner `TemporaryFiles` liegen, ist diese Option nicht verfügbar.
	Baustein bearbeiten
	Operation des Bausteins von `Read` in `Write` ändern
	Operation des Bausteins von `Write` in `Read` ändern
✕	Baustein löschen

3.2 Prozessbausteine verbinden

`Transition`: Verbindet die einzelnen Prozessbausteine und definiert die Reihenfolge der Prozessschritte durch die Pfeilrichtung.

Eigenschaften

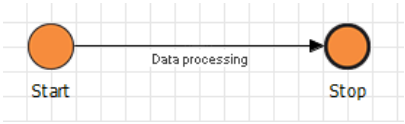
> ⚠ Einige Eigenschaften sind nur sichtbar, wenn in der Sicht **Properties** die Option **Show Advanced Properties** aktiviert ist.

`Label`	Freitextfeld zur Beschriftung des Übergangs (zur übersichtlicheren Gestaltung bzw. Dokumentation)
`Log4J Log`	Logging über *Log4j* an dieser Stelle des Prozesses mit den aktuellen Input-Daten aktivieren. Um das Logging über *Log4j* zu aktivieren, markieren Sie die Eigenschaft, und klicken Sie auf . **Mögliche Werte:** • *`Disabled:`* Keine Log4j-Aktion wurde festgelegt (Standard). • *`Enabled:`* Eine Log4j-Aktion wurde festgelegt.
`Save Point`	Save Point an dieser Stelle des Prozesses setzen, um den kompletten Zustand der Prozess-Engine mit den verarbeiteten Daten und Statusinformationen bei der Ausführung einer Prozessinstanz im Dateisystem zu persistieren. Die Prozessinstanz kann vom letzten gespeicherten Save Point über die *Monitoring-Oberflächen* wiederaufgesetzt werden. **Mögliche Werte:** • *`No Savepoint:`* Kein Save Point wurde festgelegt (Standard) • *`Save and Continue`*: Save Point erstellen und die Prozessausführung direkt fortsetzen • *`Save and Stop`*: Save Point erstellen und die Ausführung der Prozessinstanz anhalten • *`Delete`*: aktuellen Save Point löschen
`Source Component`	Baustein, an dem die Übergangslinie beginnt (wird vom X4 Designer gesetzt)

`Target Component`	Baustein, zu dem die Übergangslinie führt (wird vom X4 Designer gesetzt)
`Priority`	Bei Gabelungen (`Fork`) und Verzweigungen (`Condition`): Position in der Ausführreihenfolge (Priorität) **Mögliche Werte:** • beliebige positive ganze Zahl • *`1`*: höchste Priorität (Standard)
`Condition`	Bei Bedingungs-Bausteinen: vom **Condition Manager** hinterlegte Ausführbedingung (wird vom **Condition Manager** gesetzt)

Beispiel

Übergang zwischen zwei Prozessbausteinen mit einer Beschriftung (`Label`):



3.3 Bausteine im Bereich „Control“

Folgende Control-Bausteine stellen die Basisfunktionen für den Ablauf von Prozessen und deren Kontrollstrukturen bereit:

Auf den folgenden Seiten finden Sie weitere Informationen zur Funktion der einzelnen Bausteine.

3.3.1 Stop

Die Grundelemente jedes Prozesses sind ein definierter Prozessbeginn (`Start`) und mindestens ein Prozessende (`Stop`).
Jeder Prozess, den Sie anlegen, besteht automatisch aus einem `Start`- und einem `Stop`-Baustein.

Der `Start`-Baustein kann nicht entfernt werden, und es ist nicht möglich, weitere `Start`-Bausteine hinzuzufügen. Deshalb enthält die Palette nur den Control-Baustein `Stop` ( ).

Eigenschaften und Parameter

`Status Code` (nur `Stop`-Baustein)	Statuscode der X4 Engine. Bei Verwendung des `Stop`-Elements als ReST-Endpunkt in einem Prozess wird der HTTP-Statuscode bei Erreichen des `Stop`-Elements ausgegeben. ⚠ Ist das Flag **Stop on error** für den Prozess aktiviert, wird im Fehlerfall der Statuscode des `Stop`-Elements ignoriert und 500 zurückgegeben. Ist **Stop on error** für den Prozess deaktiviert, wird der Statuscode des `Stop`-Elements ausgegeben. **Mögliche Werte:** • leer (Standard) (ausgegebener Statuscode: 200 = fehlerfreier Prozessdurchlauf/500 = Prozessdurchlauf mit Fehler) • beliebiger numerischer Wert (gültiger HTTPS-Statuscode, z. B. 200 = fehlerfrei/500 = mit Fehler)
`Label`	Beschriftung des Symbols im Prozessdiagramm

Input und Output

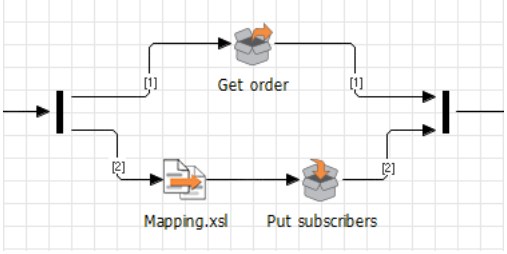
Sowohl der `Start`- als auch der `Stop`-Baustein geben einen ggf. erhaltenen Input des vorherigen Prozessschrittes (bei `Start` den Prozess-Input) ohne Veränderung an den nächsten Prozessbaustein weiter.

Der `Stop`-Baustein gibt außerdem den Status des vorherigen Prozessbausteins unverändert weiter und setzt damit den Ausführstatus des Prozesses.

3.3.2 Fork und Join

`Fork` und `Join`:

Um zwei oder mehrere voneinander unabhängige Ausführungszweige mit demselben Input auszuführen, können Sie eine Gabelung (`Fork`-Baustein) verwenden. Wieder zusammenführen können Sie die separaten Ausführungszweige mit einem `Join`-Baustein.



Eigenschaften und Parameter

`Rotation`	Drehung der Komponente **Mögliche Werte:** • *`0`*: nicht drehen (Standard) • *`90`*: um 90° gegen den Uhrzeigersinn drehen • *`180`*: um 180° drehen • *`270`*: um 270° gegen den Uhrzeigersinn drehen

Input und Output

Der `Fork`-Baustein gibt ein Input-Dokument an alle Ausführungszweige unverändert weiter.

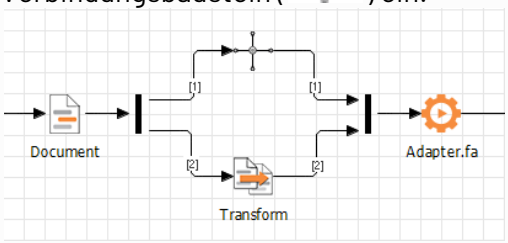
Der `Join`-Baustein hingegen behält die Output-Dokumente des letzten ausgeführten Bausteins innerhalb der Gabelung im Speicher. Wenn es sich um XML-Dokumente handelt, dann können sämtliche Output-XML-Dokumente mit einem `Merge`-Baustein in einem XML-Dokument zusammengeführt werden. Wenn die Gabelungen jedoch beispielsweise als letzten Baustein vor dem `Join` ein Binärdokument und ein XSL-Mapping zur dynamischen Parametrierung enthalten, dann kann ein Transfer-Adapter sowohl die dynamischen Parameter als auch den Binärdatenstrom verarbeiten.

Fork und Join verwenden

Ein Ausführungszweig kann aus einem oder mehreren Prozessschritten bzw. Bausteinen bestehen.

Die Ausführungszweige innerhalb einer Gabelung werden nicht parallel abgearbeitet, sondern anhand einer Priorität gesteuert, die an den Übergängen durch eine Zahl dargestellt wird. Die Priorität der Ausführungszweige können Sie in den Eigenschaften des Übergangs festlegen.

Setzen Sie bei `Fork/Join`-Konstrukten, in denen in einem Ausführzweig eine Operation ausgeführt wird (z. B. ein XSL-Mapping zur dynamischen Parametrierung eines Adapters) und im anderen Zweig lediglich das Input-Dokument durchgeleitet werden soll, einen Verbindungsbaustein ( ) ein.



ACHTUNG: Unerwartetes Verhalten der Prozess-Engine!
Wenn Sie innerhalb einer Schleife, die ein `Fork`- und `Join`-Element enthält, eine weitere Schleife in Ihrem Prozessdiagramm definieren, dann kann die Prozess-Engine den Status für die bereits abgearbeiteten Prozessschritte nicht korrekt zurücksetzen.
Lagern Sie stattdessen die innere Schleife mit sämtlichen Prozessbausteinen als Subprozess aus. Weitere Informationen hierzu finden Sie unter Subprocess.

3.3.3 Merge

`Merge`: Führt XML-Dokumente aus unterschiedlichen Quellen zu einem einzigen XML-Dokument zusammen oder führt Text- oder Binärdatenströme zusammen.

Eigenschaften

`Label`	Beschriftung des Symbols im Prozessdiagramm
`Root Element`	Name des Wurzelelements, das den gesamten Inhalt der zusammengeführten XML-Dokumente enthält. **Mögliche Werte:** • Zeichenkette, die für XML-Elementnamen gültig ist. Standard: *`Merge`*

Daten mit Merge zusammenführen

Der Merge-Baustein wird meist hinter der Zusammenführung einer Gabelung (`Join`) verwendet. Wenn XML-Dokumente zusammengeführt werden, dann enthält das durch den `Merge`-Baustein erzeugte XML-Dokument ein Wurzelelement, das wiederum die Wurzelelemente und sämtliche Knoten der zusammengeführten XML-Dokumente enthält. Diese XML-Struktur kann vom nächsten Prozessschritt verarbeitet werden. Der Name des Wurzelelements kann über die Eigenschaft `Root Element` des `Merge`-Bausteins festgelegt werden.

Hinweis:
Wenn über einen Merge-Baustein XML-Dokumente und mindestens ein Nicht-XML-Dokument zusammengeführt werden, wird die Ausgabe aller Dokumente in dem durch den `Merge`-Baust ein erzeugten Output verkettet.

Wenn zwei XML-Dokumente zusammengeführt werden, die bereits in einem vorherigen Schritt durch den `Merge`-Baustein zusammengeführt wurden, enthält das Ergebnis-XML-Dokument lediglich ein Wurzelelement und sämtliche Knoten der zusammengeführten XML-Dokumente.

Zusammenführung von zwei Dokumenten

```
<?xml version="1.0" encoding="UTF-8" ?>
<Merge>
   <!-- Daten 1 -->
   <!-- Daten 2 -->
</Merge>
```

Wenn die Input-Dokumente Textdaten (MIME-Typ `text ...` oder `application/xml`) mit derselben Zeichenkodierung sind, dann wird deren Zeichenkodierung ausgegeben. Bei unterschiedlicher Zeichenkodierung wird eine UTF-8-Sequenz ausgegeben. Wenn der MIME-Typ bei allen Input-Dokumenten gleich ist, dann hat das Ergebnis auch diesen MIME-Typ, ansonsten `text/plain`.

3.3.4 Condition

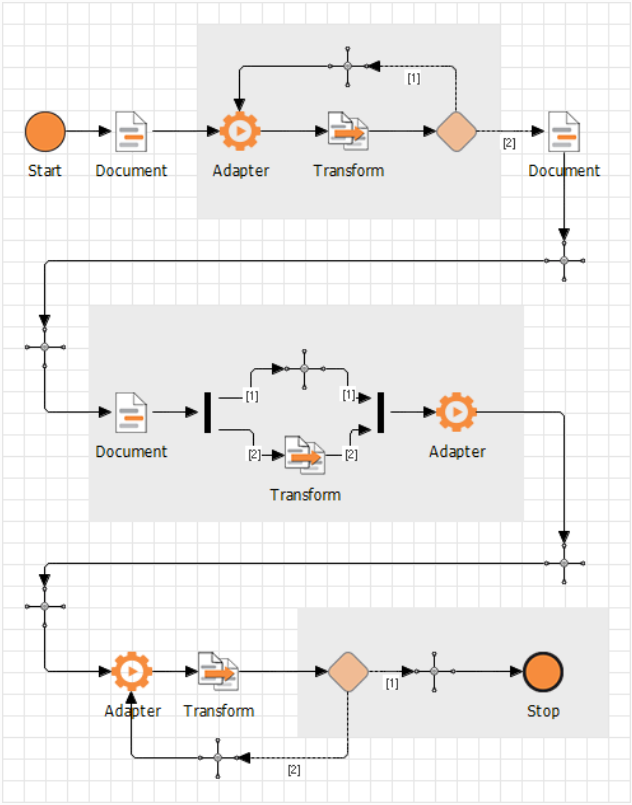
`Condition`: Trifft Fallunterscheidungen mithilfe von Verzweigungen im Prozess, wobei jeder dahinterliegende Zweig bei einer zutreffenden Bedingung ausgeführt wird.

Eigenschaften

`Label`	Freitextfeld zur Beschriftung des `Condition`-Bausteins
`Condition`	Ausführbedingung (wird automatisch gesetzt). Doppelklicken Sie auf die Verbindungslinie, um Bedingungen zu hinterlegen.

`Rotation`	Drehung des `Condition`-Bausteins auf der Zeichnungsfläche in Grad, um je nach Anordnung der Prozessbausteine (auch von rechts nach links oder von unten nach oben) den Ablauf korrekt darzustellen **Mögliche Werte:** • *`0`*: keine Drehung (Standard) • *`90`*: um 90° gedreht • *`180`*: um 180° gedreht • *`270`*: um 270° gedreht

Verwenden Sie in Schleifen beim Rücksprung vom Bedingungsbaustein zum ersten Schleifenbaustein sowie zwischen einem Bedingungsbaustein und einem direkt darauf folgenden `Stop`-Baustein einen Verbindungsbaustein (`Connection`).



3.3.4.1 Bedingungen erstellen und verwenden

Mit dem `Condition`-Baustein können Sie Verzweigungen im Prozess abbilden, wobei jeder Ausführungszweig eine Bedingung enthalten muss.
Für Fallunterscheidungen können Sie Bedingungen definieren anhand von:

- Ausführstatus (Statuswert) eines bereits ausgeführten Prozessschritts
- Parametern (globalen, System- oder Projekt-Parametern)
- Kontextvariablen

Alle Werte werden zur Prozesslaufzeit vom X4 Server ausgewertet.

Wenn Sie für einen Ausführungszweig keine Bedingung festgelegt haben, wird dieser *immer* ausgeführt. Dies wird durch ein halbtransparentes Symbol des `Condition`-Bausteins dargestellt. Wenn mehrere Bedingungen zutreffen, werden alle entsprechenden Zweige gemäß deren Priorität nacheinander ausgeführt.

Voraussetzungen
Um den `Condition`-Baustein korrekt nutzen zu können, müssen folgende Voraussetzungen erfüllt sein:

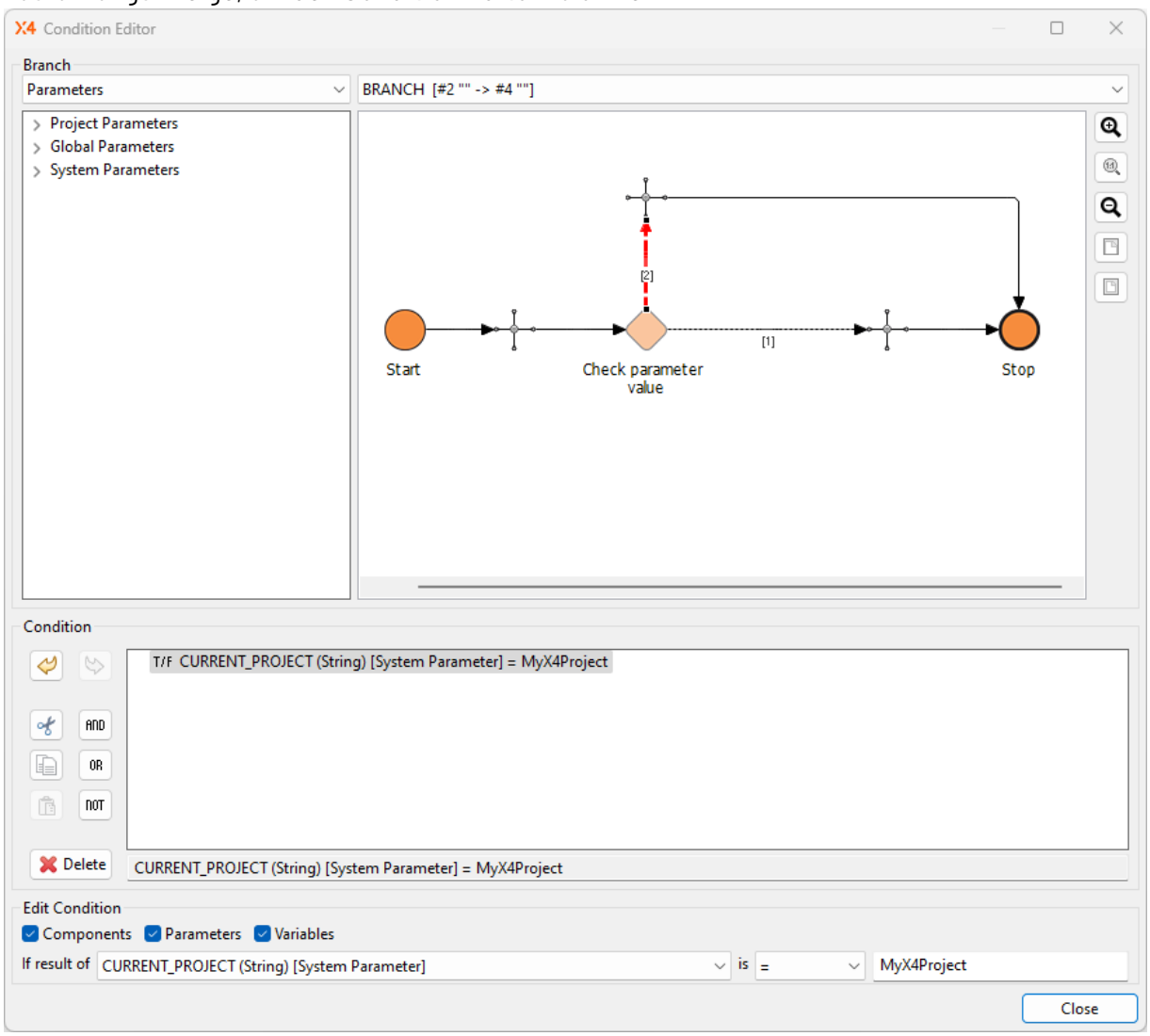
- Ein `Condition`-Baustein muss mindestens zwei Ausführungszweige besitzen.
- Ausführungszweige dürfen nicht komplett leer sein. Mindestens ein Ausführungszweig muss einen Prozessbaustein enthalten.
- Für jeden Ausführungszweig muss mindestens eine eindeutige und gültige Bedingung definiert werden.
- Der `Condition`-Baustein sollte sich nicht innerhalb eines `Fork`/`Join`-Konstruktes befinden.

3.3.4.1.1 Condition-Baustein in einen Prozess einfügen

Grundsätzlich gehen Sie wie folgt vor, um den `Condition`-Baustein in einen Prozess einzufügen:

1. Ziehen Sie den `Condition`-Baustein per Drag&Drop aus der Palette in die Zeichnungsfläche.
2. Definieren Sie Übergangslinien vom davor liegenden Prozessbaustein zum `Condition`-Baustein und für die Ausführungszweige.

3. Doppelklicken Sie auf den `Condition`-Baustein oder einen der dahinter liegenden Ausführungszweige, um den **Condition Editor** zu öffnen.



4. Legen Sie im **Condition Editor** mindestens eine Bedingung für jeden Ausführungszweig fest. Klicken Sie dazu auf einen Übergang hinter dem `Condition`-Baustein, um einen Ausführungszweig zu wählen. Der gewählte Übergang wird nun rot gestrichelt hervorgehoben. Sie können im **Condition**-Bereich jetzt Auswertungskriterien wählen (anhand des Ausführstatus, anhand von Parametern (globalen, System- oder Projekt-Parametern) sowie anhand von Kontextvariablen, siehe unten).

> ⓘ Jede Bedingung kann aus mehreren Bedingungselementen bestehen, die Sie über logische Operatoren (`AND`, `OR` und `NOT`) verknüpfen bzw. kombinieren und verschachteln können. Weitere Bedingungselemente können Sie über die entsprechenden Schaltflächen einfügen.

5. Klicken Sie auf **Close**, um die Bedingungen zu speichern.

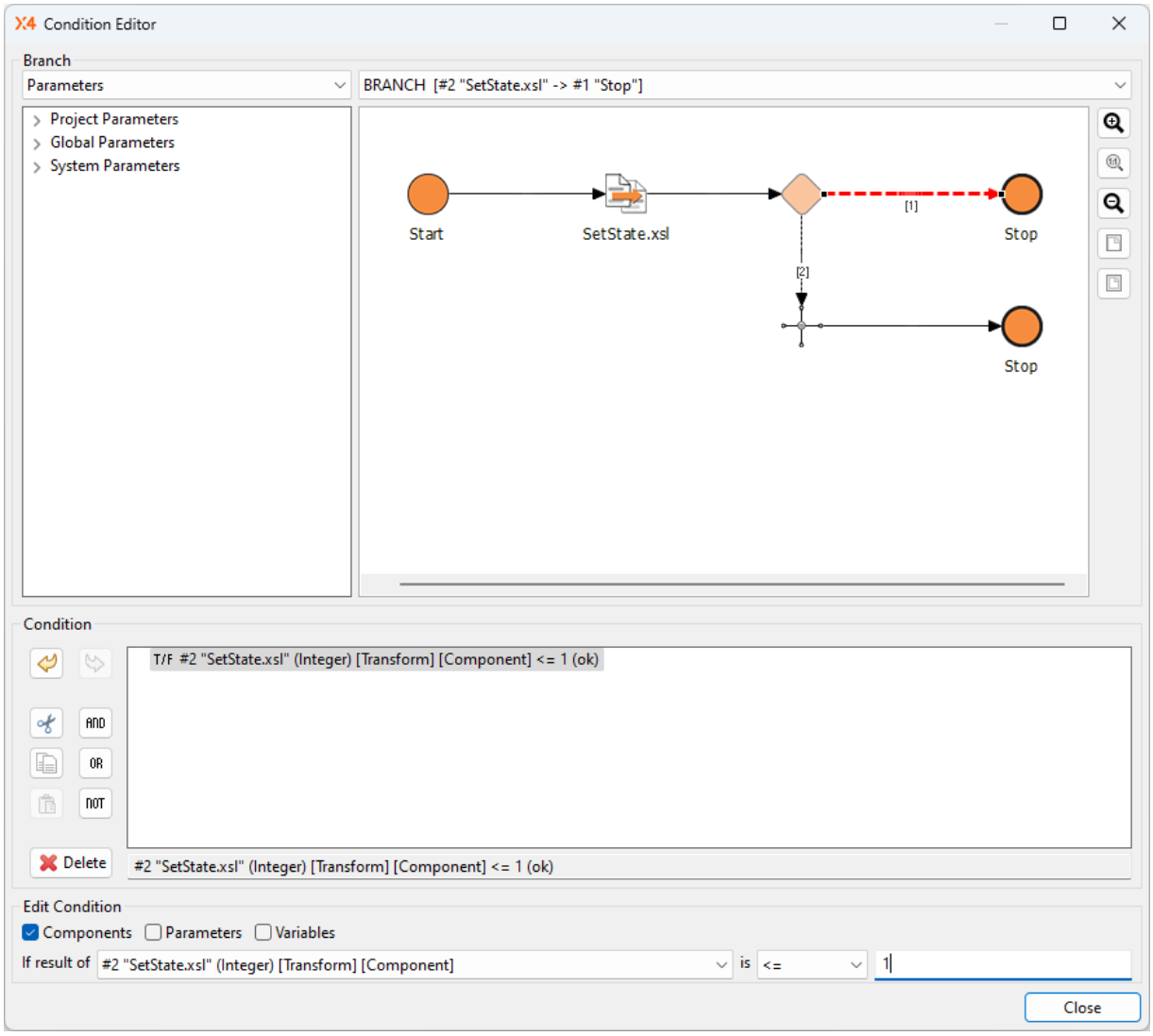
ACHTUNG: Unerwartetes Verhalten der Prozess-Engine!
Wenn Sie innerhalb einer Schleife, die ein `Fork`- und `Join`-Element enthält, eine weitere Schleife in Ihrem Prozessdiagramm definieren, dann kann der X4 Server den Status für die bereits abgearbeiteten Prozessschritte nicht korrekt zurücksetzen.
Lagern Sie stattdessen die innere Schleife mit sämtlichen Prozessbausteinen als Subprozess aus. Weitere Informationen hierzu finden Sie unter Subprocess.

3.3.4.1.2 Bedingungen anhand des Ausführstatus definieren

Mit `Condition`-Bausteinen können Sie anhand des *Statuswerts* eines bereits ausgeführten Prozessschritts eine oder mehrere Bedingungen definieren. So kann beispielsweise überprüft werden, ob ein Iterator-Adapter keine Daten mehr liefert, sodass die Prozessausführung auf einen `Stop`-Baustein umgeleitet und damit beendet wird.

Um eine Bedingung anhand des Ausführstatus zu definieren, gehen Sie wie folgt vor:

1. Klicken Sie im **Condition Editor** auf eine Übergangslinie hinter dem `Condition`-Baustein, um einen Ausführungszweig zu wählen.
2. Wählen Sie unter **Edit Condition** den Prozessbaustein aus, dessen Ausführstatus zur Prozesslaufzeit ausgewertet werden soll.
   Hier stehen nur vorangegangene Bausteine des Prozesses zur Auswahl. Alternativ können Sie auch einen Prozessbaustein aus der Prozessvorschau per Klick wählen.
3. Wählen Sie einen Vergleichsoperator aus (z. B. **<**, **=**), und geben Sie einen ganzzahligen Vergleichswert an, auf den in der Bedingung geprüft werden soll, z. B. *`1 (ok)`*.



Bei Bedarf können Sie auch Bedingungen mit logischen Operatoren (`AND`, `OR`, `NOT`) verknüpfen bzw. verschachteln.

4. Klicken Sie auf **Close**, um die Bedingungen speichern.

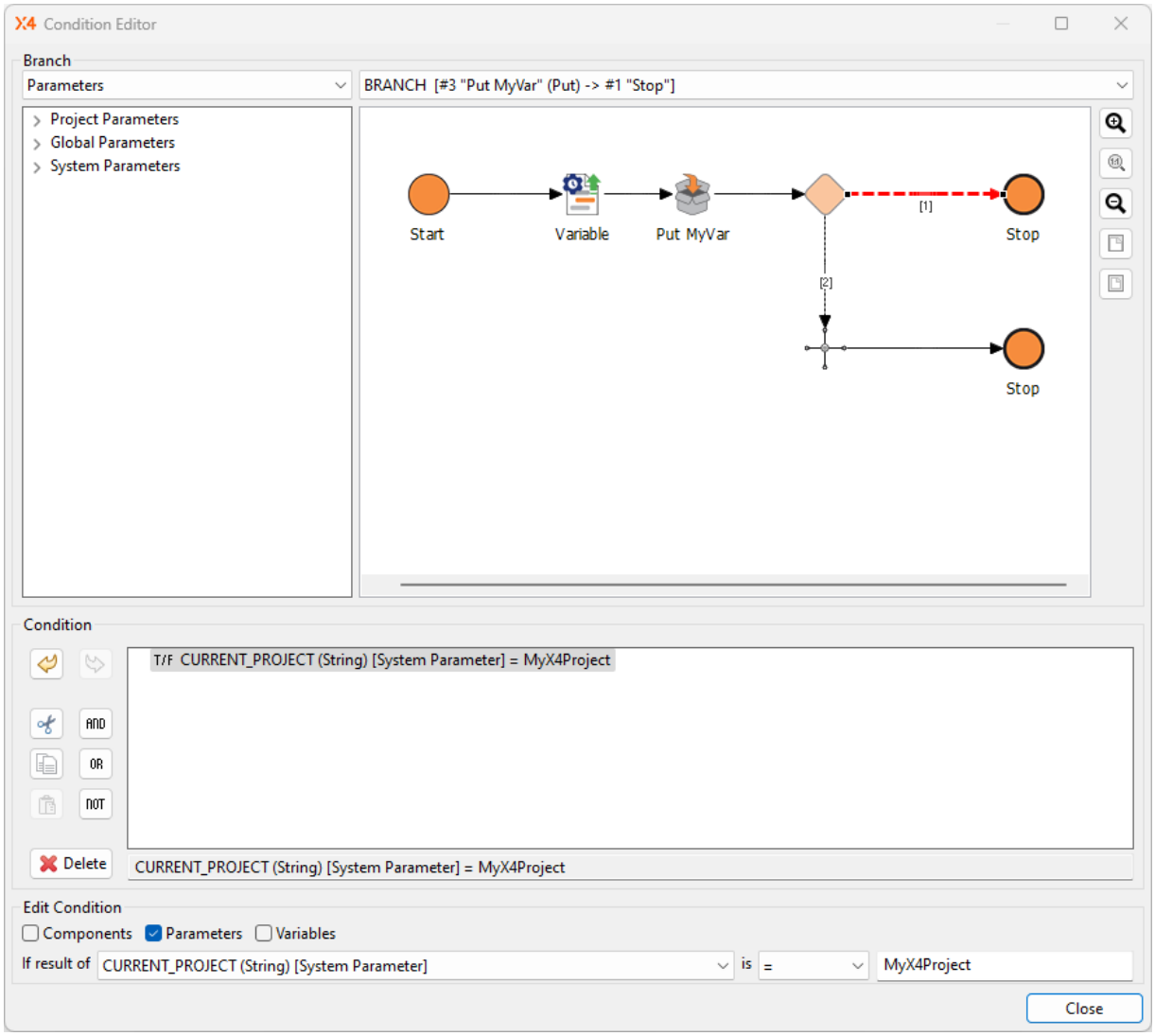
3.3.4.1.3 Bedingungen anhand von System-Parametern definieren

Mit `Condition`-Bausteinen können Sie anhand von aktuellen Umgebungsbedingungen die Ausführung des Technical Processes beeinflussen. Dazu können Sie in Bedingungen über System-Parameter wie `%CURRENT_PROJECT%` die aktuellen Werte zur Prozesslaufzeit ermitteln. Weitere Informationen hierzu finden Sie unter Übersicht der System-Parameter.

Um eine Bedingung anhand eines Parameters zu definieren, gehen Sie wie folgt vor:

1. Klicken Sie im **Condition Editor** auf eine Übergangslinie hinter dem `Condition`-Baustein, um einen Ausführungszweig zu wählen.
2. Wählen Sie in der Auswahlliste links oben den Eintrag **Parameters**, um darunter in einer Liste die verfügbaren Parametergruppen und deren Parameter anzuzeigen.

3. Wählen Sie den Parameter aus, dessen aktueller Wert zur Prozesslaufzeit ausgewertet werden soll, z. B. *CURRENT_PROJECT.*
4. Wählen Sie einen der folgenden Vergleichsoperatoren:
    - = gleich wie ...
    - `!=` ungleich ...
    - `contains` enthält ...
    - `startsWith` beginnt mit ...
    - `endsWith` endet mit ...
5. Geben Sie einen beliebigen alphanumerischen Vergleichswert an, auf den geprüft werden soll, z. B. *MyX4Project*.



Bei Bedarf können Sie auch Bedingungen mit logischen Operatoren (`AND`, `OR`, `NOT`) verknüpfen bzw. verschachteln.

6. Klicken Sie auf **Close**, um die hinterlegte Bedingung zu speichern.

3.3.4.1.4 Bedingungen anhand von globalen oder Projekt-Parametern definieren

Mit `Condition`-Bausteinen können Sie unter bestimmen Voraussetzungen auch globale oder Projekt-Parameter für die Definition von Bedingungen verwenden. Wie mit System-Parametern lässt sich somit je nach aktuellen Umgebungsbedingungen die Ausführung des Technical Processes beeinflussen.

Um in Bedingungen globale oder Projekt-Parameter im **Condition Editor** zu verwenden und zur Prozesslaufzeit auswerten zu können, gehen Sie grundsätzlich wie bei der Definition von Bedingungen mit System-Parametern vor.

3.3.4.1.4.1 Bedingungen anhand von globalen Parametern definieren

Beachten Sie jedoch folgende zusätzlichen Voraussetzungen bei globalen Parametern:

- Der globale Parameter muss bereits in der **Global Parameter Configuration** definiert sein, um im **Condition Editor** ausgewählt werden zu können. Weitere Informationen hierzu finden Sie unter Globale Parameter und -gruppen erstellen.
- Der globale Parameter muss zudem auch während der Prozessausführung in der **Global Parameter Configuration** definiert sein, um zur Laufzeit vom X4 Servers ausgewertet werden zu können.
- In den Prozess-Eigenschaften muss in der Eigenschaft `Parameter Groups` die entsprechende Parametergruppe ausgewählt sein. Weitere Informationen hierzu finden Sie unter Globale Parameter und -gruppen erstellen.

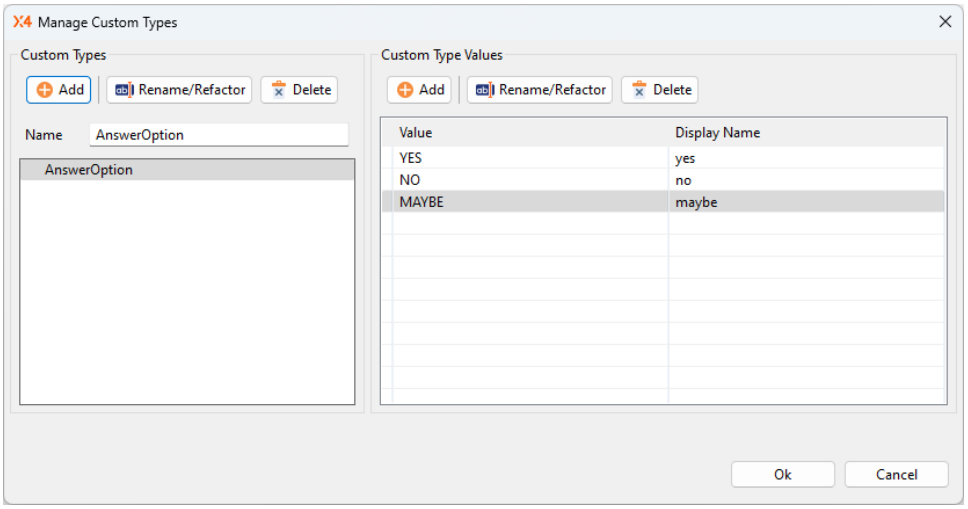
3.3.4.1.4.2 Bedingungen anhand von Projekt-Parametern definieren

Beachten Sie jedoch folgende zusätzlichen Voraussetzungen bei Projekt-Parametern:

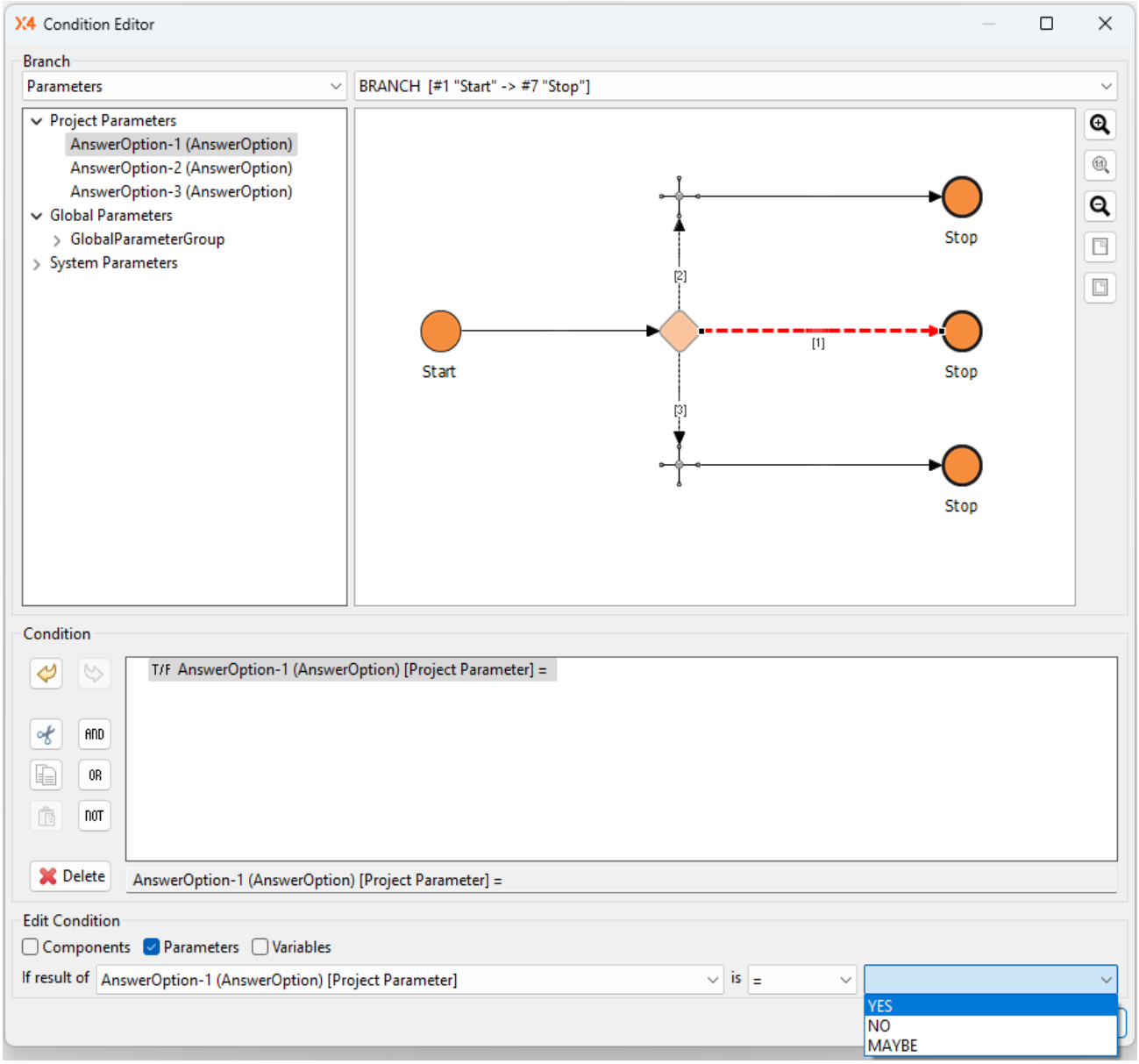
- Der Projekt-Parameter muss bereits in der Datei mit der Endung *.param im Projekt definiert sein, um im **Condition Editor** ausgewählt werden zu können. Weitere Informationen hierzu finden Sie unter Projekt-Parameter.
- Mit **Custom Types** und **Custom Type Values** können Auswahloptionen für die Verwendung im Condition Editor definiert werden. Weitere Information hierzu finden Sie unter Custom Types erstellen.

3.3.4.1.4.3 Anwendungsbeispiel für Projekt-Parameter mit Custom Types

Der **Custom Type** `AnswerOptions` hat die Werte (**Values**) `YES`, `NO`, `MAYBE` und Anzeigenamen (**Display Names**) `yes`, `no`, `maybe`:



Die **Values** sind dann im **Condition Editor** auswählbar:



3.3.4.1.5 Bedingungen anhand von Kontextvariablen definieren

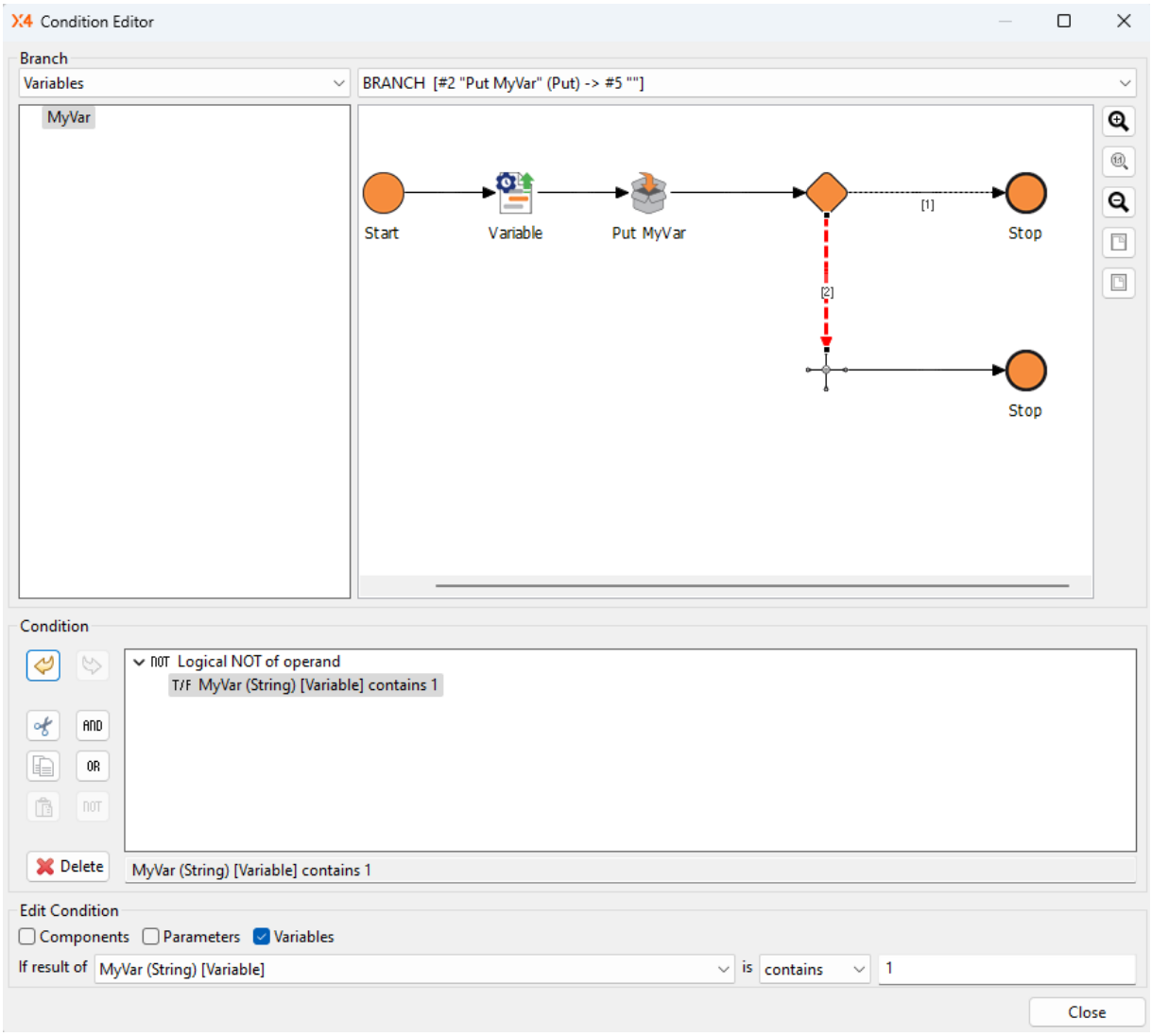
Mit `Condition`-Bausteinen können Sie anhand von aktuellen Kontextvariablenwerten Bedingungen definieren, welche die Ausführung des Technical Processes beeinflussen.

Um eine Bedingung anhand einer Kontextvariablen zu definieren, gehen Sie wie folgt vor:

1. Klicken Sie im **Condition Editor** auf eine Übergangslinie hinter dem `Condition`-Baustein, um einen Ausführungszweig zu wählen.
2. Wählen Sie in der Auswahlliste links oben den Eintrag **Variables**, um darunter eine Liste von Kontextvariablen anzuzeigen, die bereits im Prozess definiert wurden.
3. Wählen Sie die Kontextvariable aus, deren aktueller Wert zur Prozesslaufzeit ausgewertet werden soll, z. B. *MyVar*.
4. Wählen Sie einen der folgenden Vergleichsoperatoren:

- = gleich wie ...
- `!=` ungleich ...
- `contains` enthält ...
- `startsWith` beginnt mit ...
- `endsWith` endet mit ...

5. Geben Sie einen beliebigen alphanumerischen Vergleichswert an, auf den geprüft werden soll, z. B. *1*.



Bei Bedarf können Sie auch Bedingungen mit logischen Operatoren (`AND`, `OR`, `NOT`) verknüpfen bzw. verschachteln.

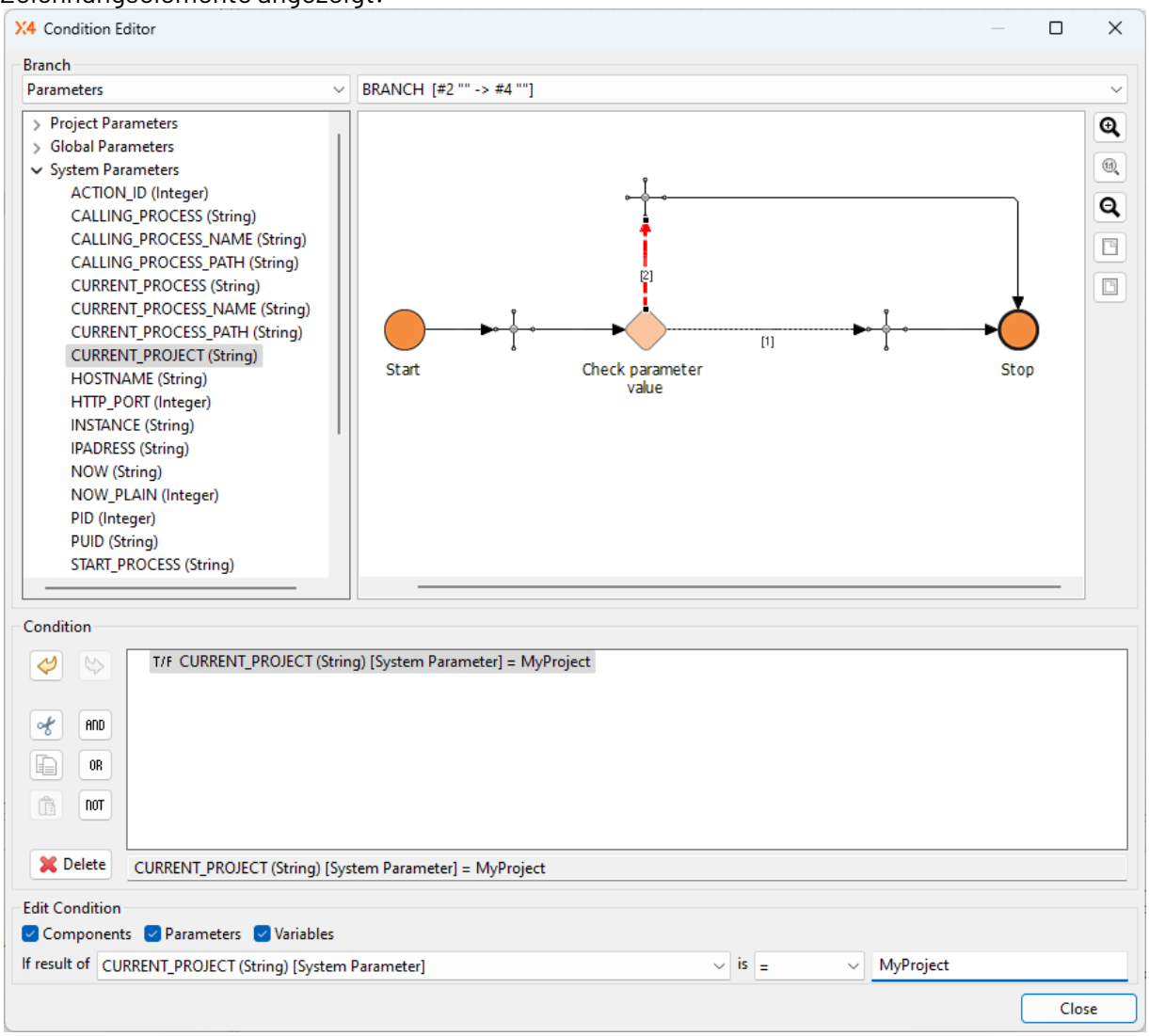
6. Klicken Sie auf **Close**, um die hinterlegte Bedingung zu speichern.

3.3.4.2 Condition Editor

Mit dem **Condition Editor** können Sie für `Condition`-Bausteine innerhalb eines Technical Processes festlegen, wann verschiedene Verzweigungen ausgeführt werden sollen. Für jede Verzweigung lassen sich Bedingungen definieren, die Statuswerte von vorangehenden Prozessbausteinen, Kontextvariablenwerte oder Parameterwerte (globale, Projekt- oder System-Parameter) zur Prozesslaufzeit auswerten.

So öffnen Sie den Condition Editor

1. Öffnen Sie einen Technical Process.
2. Doppelklicken Sie im Technical Process auf ein *Condition*-Symbol oder auf einen der Ausführungszweige hinter dem `Condition`-Baustein.
   Im **Condition Editor** wird eine vereinfachte Ansicht des Prozessdiagramms ohne Zeichnungselemente angezeigt:



Rechte Seite des Bereichs Branch

Im Bereich **Branch** wählen Sie zunächst auf der rechten Seite über die Auswahlliste den Zweig, für den Sie eine Bedingung festlegen möchten. Alternativ können Sie in der Prozessansicht auch die gewünschte Übergangslinie anklicken, um für diesen Zweig eine Bedingung zu definieren.

Ansichtsoptionen für die Prozessdarstellung:

	Zoom in	Ansicht vergrößern

	Original zoom	Ursprüngliche Größe anzeigen
	Zoom out	Ansicht verkleinern
	Zoom to page	Alles anzeigen
	Zoom to width	Ansicht auf voller Breite

Linke Seite des Bereichs Branch

Auf der linken Seite des Bereichs **Branch** können Sie über eine Auswahlliste zusätzliche Kriterien zur Definition von Bedingungen wählen. Im Bereich darunter werden gemäß Ihrer Auswahl alle verfügbaren Einträge angezeigt. Als Kriterien stehen folgende Optionen zur Auswahl:

- **Parameters**: eine Liste der aktuell verfügbaren Parametergruppen und deren Parameter (System-, Projekt- oder globale Parameter)

  > Globale Parameter sind nur auswählbar, wenn sie bereits in der Global Parameter Configuration angelegt wurden.
  > Projekt-Parameter sind nur auswählbar, wenn sie bereits in der Datei mit der Endung `*.param` des Projekts angelegt wurden (siehe Projekt-Parameter erstellen).

- **Variables**: eine Liste der Kontextvariablen, die im aktuellen Technical Process bereits definiert wurden

Bereich Condition

Im Bereich **Condition** werden die Bedingungen für den ausgewählten Zweig angezeigt und bearbeitet.

Um eine neue Bedingung zu erstellen oder zu bearbeiten, doppelklicken Sie in den Bereich **Condition**.

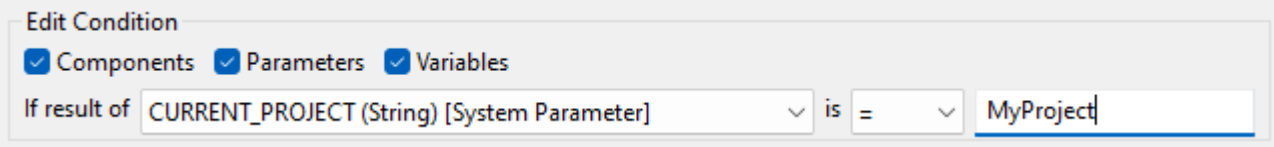
Bearbeitungsoptionen für Bedingungen:

	Undo	Rückgängig machen
	Redo	Rückgängig gemachten Schritt wiederherstellen
	Cut	Auswahl ausschneiden
	Copy	Auswahl kopieren

	Paste	Auswahl einfügen
AND	**Logical AND**	Logische UND-Verknüpfung einfügen
OR	**Logical OR**	Logische ODER-Verknüpfung einfügen
NOT	**Logical NOT**	Logische NICHT-Verknüpfung einfügen
Delete	**Delete**	Markierte Bedingung löschen

Bereich Edit Condition

Im Bereich **Edit Condition** können Sie die gewählte Bedingung erstellen oder bearbeiten. Weitere Informationen hierzu finden Sie unter Bedingungen erstellen und verwenden.



Components	Option, mit der Sie angeben können, ob in der Auswahlliste **If result of** Prozessbausteine angezeigt werden sollen, um deren Ausführstatus zur Prozesslaufzeit auszuwerten.
Parameters	Option, mit der Sie angeben können, ob in der Auswahlliste **If result of** eine Liste aller aktuell verfügbaren Parameter (System-, Projekt- und globale Parameter) angezeigt werden soll, um deren Werte zur Prozesslaufzeit auszuwerten.
Variables	Option, mit der Sie angeben können, ob in der Auswahlliste **If result of** eine Liste aller im aktuellen Prozess bereits definierten Kontextvariablen angezeigt werden soll, um deren Variableninhalt zur Prozesslaufzeit auszuwerten.
If result of	In dieser Auswahlliste wählen Sie das Kriterium (Prozessbaustein, Parameter oder Variable) für Ihre Bedingung.

is	Hier wählen Sie einen Vergleichsoperator: • < kleiner als ... • <= kleiner als oder gleich wie ... • = gleich wie ... • >= größer als oder gleich wie ... • > größer als ... • <> ungleich zu ... Für Parameter und Variablen stehen hier folgende Vergleichsoperatoren für Zeichenketten zur Verfügung: • = gleich wie ... • `!=` ungleich ... • `contains` enthält ... • `startsWith` beginnt mit ... • `endsWith` endet mit ...
(Vergleichswert)	Im Eingabefeld geben Sie den Wert an, der mit dem ausgewählten Kriterium verglichen werden soll. a) Bei Bezug auf Prozessbausteine ist dies immer ein numerischer Statuswert: • `1`: Standardwert bei erfolgreicher Ausführung • `-1`: Standardwert bei fehlerhafter Ausführung • `0`: Standardwert bei leerem Ergebnis bzw. noch nicht erreichter erfolgreicher Ausführung • `nnnn`: beliebiger ganzzahliger Wert (z. B. 2000) bei Bezug auf ein ConditionState-Mapping b) Bei Bezug auf Kontextvariablen und Parameter ist dies ein alphanumerischer Vergleichswert. • Inhalt einer zuvor mit `Put` erzeugten Kontextvariable • Inhalt eines Parameters (System-Parameters oder zuvor definierten globalen oder Projekt-Parameters)

3.3.5 Connection

`Connection`: Gibt den Input aus dem vorangegangenen Prozessschritt ebenso wie dessen Status unverändert aus. Es markiert den jeweiligen Ausführungszweig für den X4 Server eindeutig. So lässt sich mit diesem Baustein in Schleifen der Rücksprung zum ersten Schleifenbaustein explizit setzen. Er verbindet einen `Condition`-Baustein mit einem `Stop`-Baustein oder einen `Fork`-Baustein mit einem direkt darauf folgenden `Join`-Baustein.

Eigenschaft

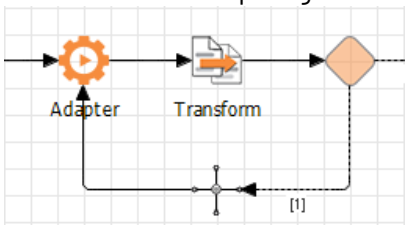
`Label`	Beschriftung des Symbols im Prozessdiagramm

Output

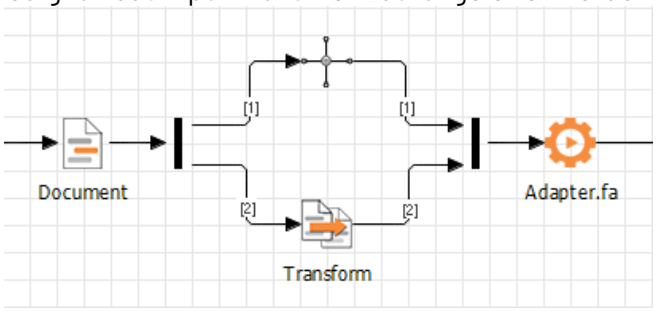
Der Verbindungsbaustein gibt den Input unverändert aus, inklusive Ausführstatus des vorherigen Prozessbausteins.

Beispiele

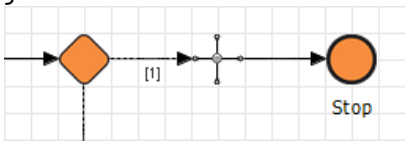
- Bei Schleifenrücksprüngen wird als letzter Schleifenbaustein ein Verbindungsbaustein gesetzt.



- Bei `Fork/Join`-Konstrukten, in denen in einem Ausführzweig eine Operation ausgeführt wird (z. B. ein XSL-Mapping zur dynamischen Parametrierung eines Adapters) und im anderen Zweig lediglich das Input-Dokument durchgeleitet werden soll, wird ein Verbindungsbaustein gesetzt.



- Zwischen einem `Condition`-Baustein und einem `Stop`-Baustein wird ein Verbindungsbaustein gesetzt.



3.3.6 Pause

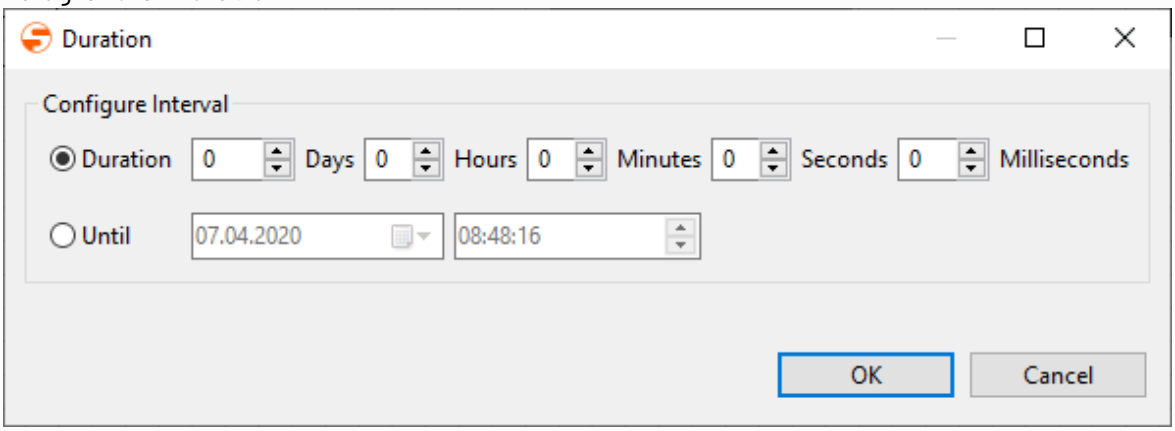
`Pause`: Definiert Wartezeiten innerhalb eines Prozesses. Nach einer festgelegten Dauer oder zu einem bestimmten Zeitpunkt wird die Ausführung des Prozesses fortgesetzt. Dabei kann ein `Pause`-Baustein einen Input vom vorangegangenen Prozessbaustein erhalten und diesen an den darauf folgenden Prozessbaustein unverändert weitergeben.

Eigenschaften

`Label`	Beschriftung des Symbols im Prozessdiagramm

`Duration`	Dauer der Pause **Mögliche Werte:** • beliebige Dauer mit Angabe von Tagen, Stunden, Minuten, Sekunden und Millisekunden • *`9999d 23h 59m 59s 999ms`*: maximale Wartezeit pro `Pause`-Baustein • *`0d 0h 0m 0s 0ms`*: keine Wartezeit (Standard) Alternativ zur Angabe einer Wartezeit lässt sich über `Until` ein Endzeitpunkt im Format `yyyy-MM-dd hh:mm:ss` für die Pause festlegen, z. B. *`2010-08-30T01:55:10`*.

Um die Pausendauer zu definieren, doppelklicken Sie auf den `Pause`-Baustein, oder klicken Sie in der Sicht **Properties** unter `Duration` auf . Anschließend definieren Sie die Dauer im Dialogfenster **Duration**.



3.3.6.1 Pause-Baustein dynamisch parametrieren

Der `Pause`-Baustein lässt sich über die technischen Parameter `sleep_time` und `sleep_until` auch dynamisch parametrieren.

- **Dynamische Parametrierung über `sleep_time`**
  Im folgenden Beispiel wird der `Pause`-Baustein auf eine Pausendauer von 5 Sekunden parametriert:

Dynamische Parametrierung über sleep_time

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
  <xsl:output method="xml" media-type="text/xml"></xsl:output>
  <xsl:template match="/">
    <!-- Pause duration 5 seconds / used format "0(d):0(h):0(m):0(s).0(ms)"-->
    <xsl:processing-instruction name="DynamicParameter">
        <xsl:text>sleep_time="0:0:0:5.0"</xsl:text>
    </xsl:processing-instruction>
    <Empty></Empty>
    <!-- <xsl:processing-instruction name="NoResult"></xsl:processing-
instruction> -->
  </xsl:template>
</xsl:stylesheet>
```

- **Dynamische Parametrierung über `sleep_until`**
  Im folgenden Beispiel wird der `Pause`-Baustein auf das aktuelle Datum und die aktuelle Uhrzeit zuzüglich 1 Minute und 15 Sekunden parametriert:

Dynamische Parametrierung über sleep_until

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:xs="http
://www.w3.org/2001/XMLSchema" version="2.0">
  <xsl:output method="xml" media-type="text/xml"></xsl:output>
  <xsl:template match="/">
    <xsl:processing-instruction name="DynamicParameter">
        <xsl:text>sleep_until="</xsl:text>
            <xsl:value-of select="format-dateTime(current-dateTime()+
xs:dayTimeDuration('PT0H1M15S'),'[Y0001]-[M01]-[D01] [H10]:[m01]:[s01]')"></xsl:
value-of>
        <xsl:text>"</xsl:text>
    </xsl:processing-instruction>
  <!-- <Empty></Empty> -->
  <xsl:processing-instruction name="NoResult"></xsl:processing-instruction>
  </xsl:template>
</xsl:stylesheet>
```

3.3.7 Counter

`Counter`: Ermöglicht die Implementierung von Zählschleifen in Prozessen. Hierfür wird in der Eigenschaft `Start` ein Anfangswert und in der Eigenschaft `End` ein Endwert gesetzt. Bei jedem Durchlauf des Zählers wird dessen Wert um die in Eigenschaft `Step Size` angegebenen Wert verringert bzw. erhöht, bis der Endwert erreicht ist. Über ein XSLT-Mapping lassen sich die Eigenschaften dynamisch setzen und der aktuelle Index kann über `indexName` in eine Variable

gespeichert werden.

Eigenschaften

`Start` Dynamischer Parameter: `start`	Anfangswert, von dem heruntergezählt bzw. hochgezählt wird **Mögliche Werte**: beliebige ganze Zahl, die größer `0` ist ⓘ Der Anfangswert wird beim Durchlauf nicht mitgezählt. Beachten Sie, dass der Startwert (`start`) lediglich beim ersten Schleifendurchlauf dynamisch parametriert werden kann.
`End` Dynamischer Parameter: `end`	Endwert, bis zu dem heruntergezählt bzw. hochgezählt wird **Mögliche Werte**: beliebige ganze Zahl, die größer `0` ist. Wenn der Endwert kleiner als der Startwert ist, wird heruntergezählt. Andernfalls wird hochgezählt. ⓘ Der Endwert wird beim Durchlauf nicht mitgezählt.
`Step Size` Dynamischer Parameter: `step`	Zählschritte für das Hochzählen bzw. Herunterzählen, z. B. `2` **Mögliche Werte**: beliebige ganze Zahl, die größer `0` ist
`Label`	Beschriftung des Symbols im Prozessdiagramm
`Index Name` Dynamischer Parameter: `indexName`	Name der Variablen **Mögliche Werte:** Name der Variablen, um auf den aktuellen Index des Zählers zuzugreifen

Input

Der Baustein erwartet nur dann einen Input, wenn die Eigenschaften dynamisch gesetzt werden.

Output

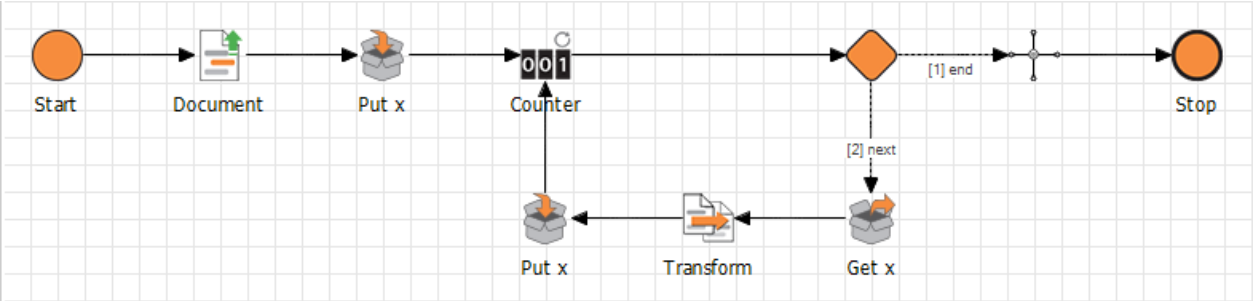
Wird die Eigenschaft dynamisch gesetzt, gibt der Baustein eine Variable mit *<indexName>* und den Wert des aktuellen Index aus. Andernfalls liefert der Baustein keinen Output.

Statuswerte

`0 (counting)`	Zähler läuft (Endwert wurde noch nicht erreicht)
`1 (finished)`	Herunterzählen ist beendet (Endwert wurde erreicht)

Beispiel

Einfache Zählschleife mit Zähler- und Bedingungs-Baustein (`Counter` und `Condition`). Dabei wurden die Eigenschaften `Start=1`, `End=4` und `Step Size=1` festgelegt. Die Zählschleife wird zweimal durchlaufen, da der Anfangswert und der Endwert nicht mitgezählt werden.



3.3.8 Transaction

`Transaction`: Führt Prozessschritte transaktionssicher durch: Bei einem *Commit* werden sämtliche Operationen auf dem Zielsystem persistent ausgeführt, bei einem *Rollback* werden alle Operationen auf den Ursprungszustand vor der Transaktion zurückgesetzt. Um Prozessschritte in eine Transaktion zu kapseln, setzen Sie vor und hinter die Prozessschritte die entsprechenden Transaktionsbausteine. Der Transaktionsbaustein wird hauptsächlich für JMS-Warteschlangen-Operationen und für Datenbankoperationen verwendet.

Beachten Sie folgende Hinweise:

- Transaktionen können nicht im Debug-Modus des X4 Designers durchgeführt werden.
- Innerhalb von Transaktionen können auch Subprozesse ausgeführt werden.
- Wenn innerhalb einer Transaktion ein Subprozess über den Process Starter-Adapter gestartet wird, dann gilt die Transaktion nicht für diesen Subprozess (Transaktionen sind nicht thread-übergreifend).
- Transaktionen in Transaktionen zu verschachteln ist nicht möglich.
- Innerhalb einer Transaktion können mehrere Datenbankverbindungen und Warteschlangen verwendet werden.
- Die verwendete Datenbankverbindung (JDBC DataSource) muss Transaktionen unterstützen.
- Um Transaktionen ausführen zu können, darf bei der Konfiguration der zu verwendenden Datasource innerhalb der `standalone.xml` das Attribut `jta` nicht auf `false` gesetzt werden.

Eigenschaften

`Label`	Beschriftung des Symbols im Prozessdiagramm

`Operation`	Transaktionsoperation festlegen **Mögliche Werte:** • *`Not assigned:`* noch keine Transaktion festgelegt • *`Begin`*: Transaktion beginnen • *`Commit`*: Transaktion durchführen/abschließen • *`Rollback`*: Transaktion abbrechen und innerhalb der Transaktion durchgeführte Operationen rückgängig machen

Statuswerte

`1`	Transaktionsbaustein wurde erfolgreich ausgeführt.
`0`	Transaktionsbaustein wurde ohne Ergebnis ausgeführt.
`-1`	Während der Transaktion trat ein Fehler auf.

3.3.9 Variable

`Variable`: Speichert temporär Daten, während ein Technical Process ausgeführt wird. Innerhalb desselben Technical Processes oder eines Subprozesses können Sie die Variable auslesen. Mithilfe des `context:`-Protokolls können Sie auch in XSL-Mappings auf solche Variablen zugreifen.

Eigenschaften

`Operation`	Bestimmt, ob die Variable initialisiert, verwendet oder verworfen werden soll **Mögliche Werte:** • *`Put`*: Daten aus dem vorangegangenen Prozessschritt in der Variable speichern (Standard) • *`Get`*: Variableninhalt lesen und an den nächsten Prozessschritt weitergeben • *`Remove`*: Variable verwerfen (aus dem Arbeitsspeicher löschen) • *`Test`*: Prüfen, ob eine Variable vorhanden ist. Ist keine Variable gesetzt, wird der Status *`0`* ausgegeben.

`Name`	Eindeutige Bezeichnung der Variable **Mögliche Werte:** • Zeichenkette, die aus alphanumerischen *US-ASCII*-Zeichen (*`Aa–Zz`* und *`0–9`*) sowie aus Unterstrichen (_) bestehen darf und nicht mit einer Ziffer beginnt (z. B. *`Report_Data01`*) • *`X4STATE`*: Enthält den Status des vorherigen Prozessschritts als XML-Dokument; siehe *Kontextvariablen.* • *`X4PREVACTION`*: Enthält Action ID, Beschriftungstext und Status des vorherigen Prozessschritts als XML-Dokument

3.3.9.1 Variablen verwenden

In Kontextvariablen lassen sich Daten im Kontext eines aktuell ausgeführten Technical Processes zwischenspeichern. Jede Kontextvariable muss zunächst initialisiert werden, also einen eindeutigen Namen erhalten und mit einem Wert belegt werden (`Put`-Operation). In späteren Prozessschritten kann dann per `Get`-Operation auf die Kontextvariable zugegriffen werden. Schließlich können Variableninhalte über die `Remove`-Operation aus dem Speicher gelöscht werden.

Variablen-Baustein erstellen	Per Drag&Drop aus der **Palette** in den Technical Process ziehen und in der Sicht **Properties** in der Eigenschaft `Name` einen eindeutigen Variablennamen eingeben (z. B. *`Var1`*)
Variable initialisieren	In der Sicht **Properties** in der Eigenschaft `Operation` den Wert *`Put`* wählen, um die Daten aus dem vorangegangenen Prozessschritt in die Variable zu laden
Variable auslesen	Den Variablen-Baustein mit demselben Namen erneut einfügen und die Eigenschaft `Operation` auf *`Get`* setzen
Variable testen	Den Variablen-Baustein mit demselben Namen erneut einfügen und die Eigenschaft `Operation` auf *Test* setzen
Variable aus dem Speicher löschen	Den Variablen-Baustein mit demselben Namen erneut einfügen und die Eigenschaft `Operation` auf *`Remove`* setzen
Innerhalb von XSL-Mappings auf Prozessvariablen zugreifen	Das `context://`-URL-Protokoll verwenden, z. B. über den XPath-Ausdruck `document(' context://Var1 ')`

3.3.9.2 Kontextvariablen

In Technical Processes stehen standardmäßig eine Reihe von Kontextvariablen zur Verfügung, die Informationen zur aktuell ausgeführten Prozessinstanz liefern. Diese Variablen können Sie sowohl im Technical Process über einen `Variable`-Baustein über die Operation `Get` als auch in XSL-Mappings über das `context:`-URL-Protokoll abrufen.

X4STATE	Gibt den Status des vorherigen Prozessschritts (unmittelbar vor dem aktuellen Prozessschritt) in einem XML-Dokument aus. **Beispiel** `<state>-1</state>`

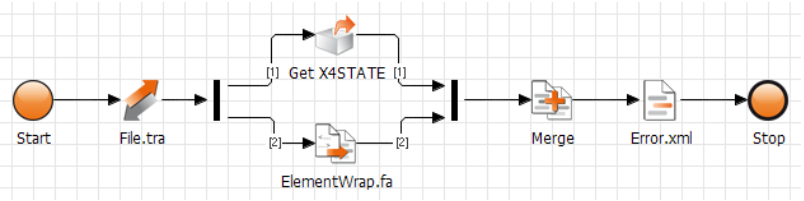
X4PREVACTION	Gibt Informationen (Action ID, Beschriftungstext und Status) zum vorherigen Prozessschritt als XML-Dokument aus.
	Beispiel
	`<action>`
	`  <id>4</id>`
	`  <label>File.tra</label>`
	`  <state>-1</state>`
	`</action>`

Beispiel

Mit diesen Kontextvariablen wird eine erweiterte Fehlerbehandlung ermöglicht. So können Sie z. B. den Fehlermeldungstext eines Adapters gemeinsam mit dem Adapterstatus ausgeben.

```
<Merge>
   <state>-1</state>
   <error>General File Error (see log for details).</error>
</Merge>
```

Adapter-Fehlermeldung und dessen Statuscode zusammengeführt in ein XML-Dokument:



3.4 Bausteine im Bereich „Modules“

Die folgenden Modul-Bausteine verbinden IT-Prozesse mit der Außenwelt und anderen Prozessen, beispielsweise in Form von Datenbankverbindungen, Funktions-Adaptern und Subprozessen. Modul-Bausteine liegen als wiederverwendbare Dateien in der Sicht **Projects** vor. Deren Konfiguration erfolgt entweder in der jeweiligen Adapter-Konfiguration oder direkt in einem XSL-Stylesheet (bei Mappings und Reports).

Auf den folgenden Seiten finden Sie weitere Informationen zur Funktion der einzelnen Bausteine.

3.4.1 Function Adapter

`Function Adapter`: Stellt grundlegende oder zusätzliche Funktionen bereit, etwa Zeichensatz-Konvertierungen oder ermöglicht beispielsweise den Zugriff auf ein SAP-System. Da der X4 Designer hauptsächlich mit XML-Daten arbeitet, werden mit Adaptern meist XML-Dokumente erzeugt oder verändert. Als Output erzeugen Adapter vorwiegend XML-Daten, die in einem nächsten Prozessschritt weiterverarbeitet werden.

> ⓘ Um den Adaptertyp zu wählen und Parameter festzulegen, doppelklicken Sie in der Zeichnungsfläche auf das entsprechende Funktions-Adapter-Symbol. Die spezifische Bedeutung der im Folgenden genannten Eigenschaften der Funktions-Adapter ist in der jeweiligen Adapter-Referenz beschrieben.

Eigenschaften

`Adapter type`	Name des Adapters
`Component Parameters`	Component-Parameter setzen, siehe Component-Parameter verwenden
`Document`	Pfad zum verknüpften Adapter im Repository
`Label`	Bezeichnung des Adapters im Technical Process (Freitext)
`Operation`	Bestimmt, welche Operation mit dem Funktions-Adapter durchgeführt wird. Je nach Adapter unterscheiden sich die Werte und Funktionen.

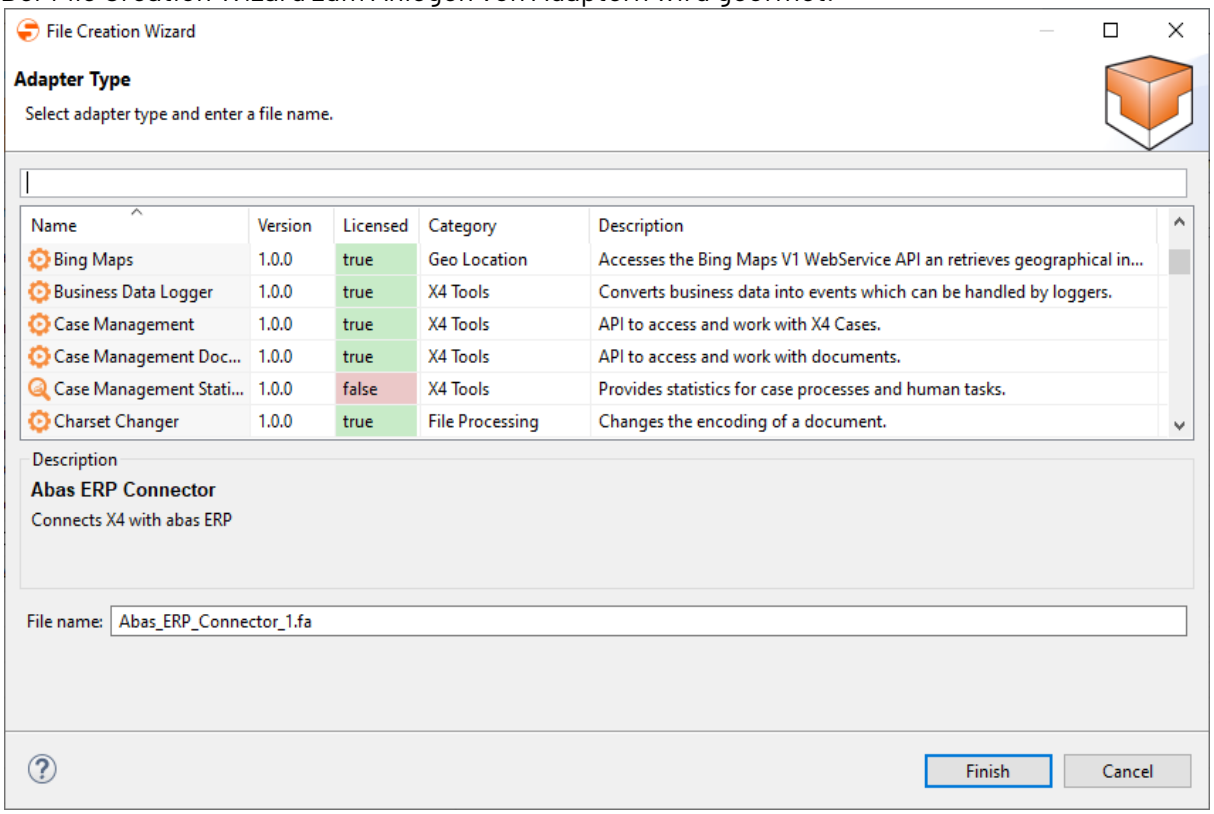
Adapter anlegen und verwenden

Adapter lassen sich innerhalb eines ESB-Projekts im Ordner `Adapters` anlegen. Hierzu stehen zwei Möglichkeiten zur Verfügung:

- über das Menü **Project > Add Adapter**
- über das Kontextmenü **New > Adapter**

Um einen Adapter anzulegen und zu parametrieren, gehen Sie wie folgt vor:

1. Klicken Sie in einem ESB-Projekt mit der rechten Maustaste auf den Ordner `Adapters`, um das Kontextmenü aufzurufen.
2. Rufen Sie im Kontextmenü **New > Adapter** auf.
   Der File Creation Wizard zum Anlegen von Adaptern wird geöffnet.

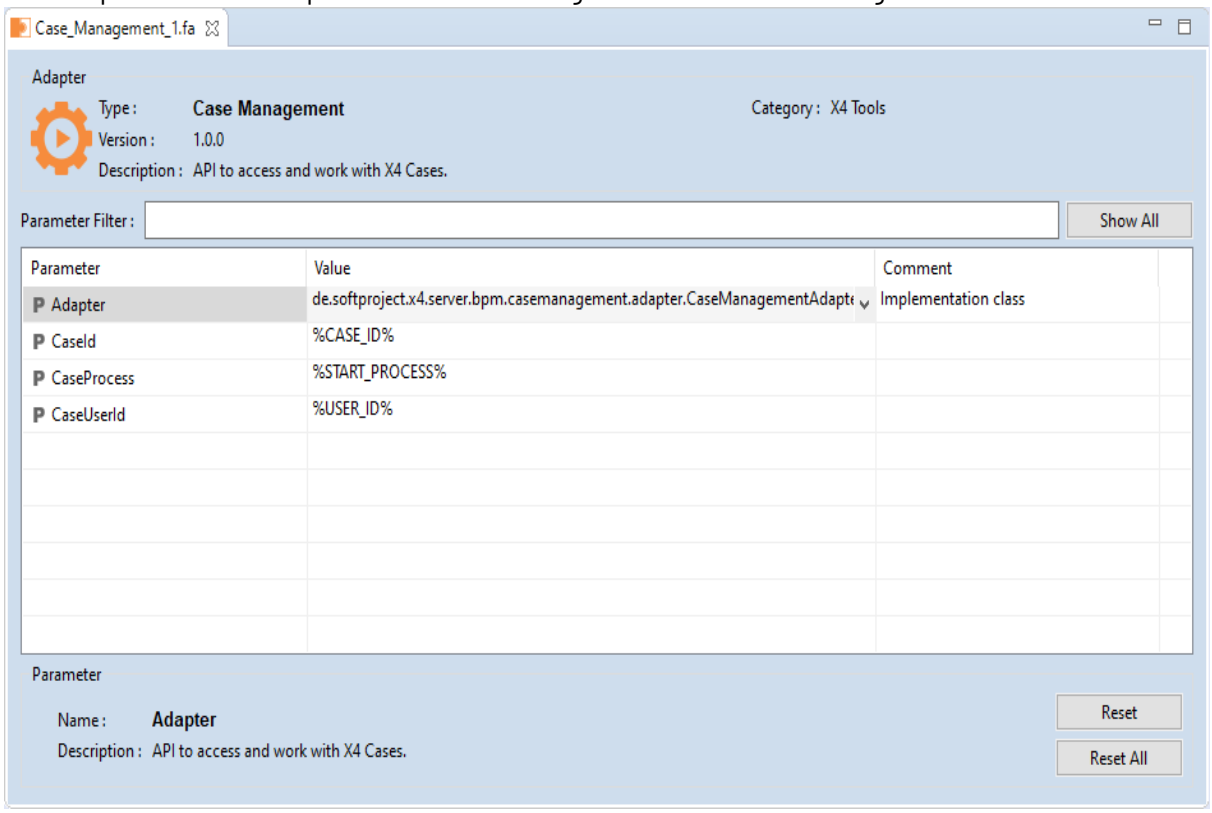


3. Wählen Sie den gewünschten Adapter aus der Liste.

   > Im Suchfeld können Sie die Adapternamen und Beschreibungen nach einem Stichwort durchsuchen und somit die angezeigten Adapter einschränken. Zudem können Sie die Adapter durch Klicken auf den Spaltentitel `Category` nach Kategorien sortieren.

4. Geben Sie in **File name** den Adapter-Namen ein.

5. Klicken Sie auf **Finish**, um den Adapter in der Sicht **Projects** anzulegen.
   Der Adapter wird im Adapter-Editor zur Konfiguration der Parameter geöffnet.



6. Hinterlegen Sie im Adapter-Editor die gewünschten Parameterwerte, mindestens jedoch die Pflichtparameter.
   Pflichtfelder erkennen Sie am Symbol
   P
   .

   - Setzen Sie alle Adapter-Parameter, die vertrauliche Informationen enthalten, immer dynamisch per XSL-Mapping, z. B. über die Processing Instruction `DynamicParameter`. Weitere Informationen hierzu finden Sie unter Benutzerdefinierte Parameter verwenden. So werden die Zugangsdaten erst zur Laufzeit des Technical Processes im Adapter gesetzt, und die entsprechenden Parameter müssen nicht im Adapter angegeben werden und sind somit nicht einsehbar.
   - Eine Beschreibung der Parameter für einzelne Adapter finden Sie unter Übersicht der X4 Adapter.

7. Speichern Sie den Adapter.
8. Fügen Sie den Adapter per Drag&Drop aus der Sicht **Projects** in den Technical Process ein.
9. Wählen Sie in der Sicht **Properties** für den markierten Adapter in der Eigenschaft `Operation` eine Adapter-Operation aus.
   Der Funktions-Adapter ist nun einsatzbereit.
10. Speichern Sie den Technical Process.

3.4.2 XML

`XML`: Bindet wohlgeformte XML-Daten in den Prozess ein. Dabei wird die Zeichenkodierung aus der XML-Deklaration und der MIME-Typ aus der Dateinamensendung ermittelt.

Eigenschaften

`Document`	Pfad zum verknüpften XML-Dokument im Repository
`Label`	Beschriftung des Symbols im Prozessdiagramm
`Operation`	Bestimmt, ob die Datei gelesen oder geschrieben werden soll **Mögliche Werte:** • *`Read`*: XML-Dokument in den Prozess einlesen und Daten an den nächsten Prozessschritt weitergeben (Standard) • *`Write`*: XML-Dokument mit den Daten aus dem vorangegangenen Prozessschritt speichern
`Type`	Bestimmt den Typ der Datei **Mögliche Werte:** • `Resource` • `Buffer` (Standard)

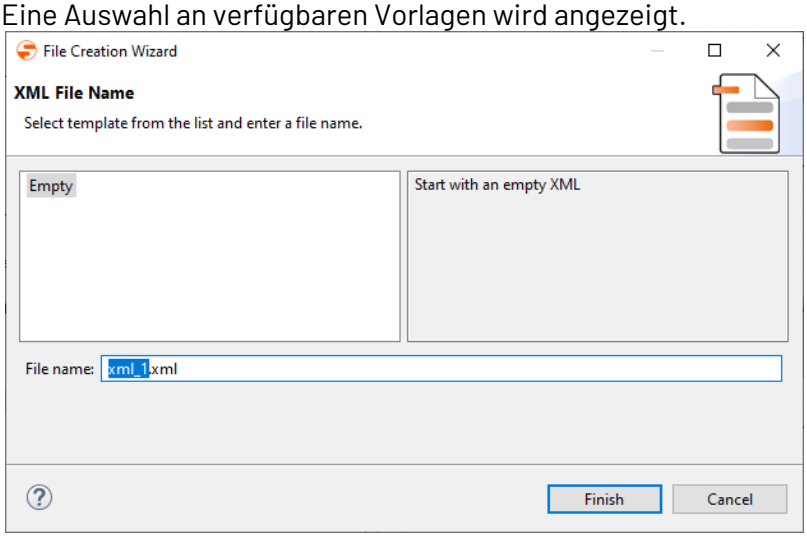
XML-Dokument erstellen und verwenden

XML-Dokumente lassen sich innerhalb eines ESB-Projekts im Ordner `Resources` (für Ressourcen) und im Ordner `TemporaryFiles` (für Puffer-Dateien) anlegen. Hierzu stehen zwei Möglichkeiten zur Verfügung:

- über das Menü **Project > Add XML Document**
- über das Kontextmenü **New > XML Document**

1. Klicken Sie im ESB-Projekt mit der rechten Maustaste auf den Ordner, um das Kontextmenü aufzurufen.

2. Rufen Sie im Kontextmenü **New> XML Document** auf.
   Eine Auswahl an verfügbaren Vorlagen wird angezeigt.



3. Wählen Sie eine leere Vorlage oder eine vorhandene Vorlage.
4. Geben Sie in **File Name** den Dateinamen ein.
5. Klicken Sie auf **Finish**, um das XML-Dokument anzulegen.
   Das XML-Dokument wurde angelegt und wird im XML Editor geöffnet.
6. Bearbeiten Sie das XML-Dokument wie gewünscht.
7. Ziehen Sie das XML-Dokument per Drag&Drop aus der Sicht **Projects** in das Prozessdiagramm.
   Beispiel: Zwischen Funktions-Adapter und `Stop`-Baustein einfügen.
8. Legen Sie in der Sicht **Properties** in `Operation` fest, ob das XML-Dokument gelesen (*`Read`*) oder geschrieben (*`Write`*) werden soll.

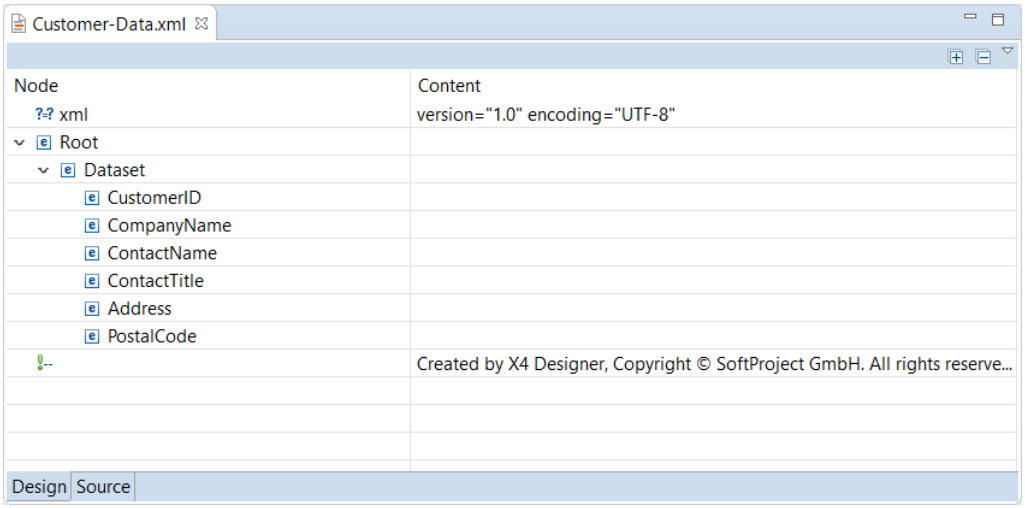
3.4.2.1 XML Editor

Mit dem XML Editor können Sie wohlgeformte XML-Dokumente erstellen und bearbeiten.

> Um den XML Editor zu starten, legen Sie entweder ein neues XML-Dokument an oder öffnen ein bereits vorhandenes Dokument.

Aufbau des XML-Editors

- Zum Bearbeiten von XML-Dokumenten stehen zwei Sichten zur Verfügung: die Sicht **Design** und die Sicht **Source**. Über die Registerkarten am unteren Rand des Editors können Sie zwischen den Sichten wechseln.



Design	In der Sicht **Design** wird die Baumstruktur des geöffneten XML-Dokuments tabellarisch dargestellt. In der linken Spalte befinden sich die verschiedenen Knoten, die Sie auf- und zuklappen können, sofern Kindknoten existieren. In der rechten Spalte werden die Inhalte der Knoten angezeigt. Über das Kontextmenü können Sie die folgenden Aktionen ausführen: • **Remove**: Markierten Knoten entfernen • **Add DTD Information**: Strukturdefinition in Form einer DTD (*Document Type Definition*) hinzufügen • **Edit Namespaces**: Namensräume bearbeiten • **Add Attribute**: Neues Attribut hinzufügen • **Add Child**: Neuen Kindknoten (Element, Processing Instruction usw.) hinzufügen • **Add Before**: Neuen Knoten vor dem markierten Knoten einfügen • **Add After**: Neuen Knoten hinter dem markierten Knoten einfügen
Source	In der Sicht **Source** können Sie den Quelltext von XML-Dokumenten bearbeiten. Dabei unterstützt Sie der Editor durch folgende Funktionen: • Syntax-Highlighting: Die verschiedenen Knotentypen werden farblich hervorgehoben. • Wohlgeformtheit: Beim Speichern erfolgt eine Prüfung auf Wohlgeformtheit des XML-Dokuments. • Eingabehilfe: Der Editor macht Vorschläge, wenn Sie neue Elemente einfügen möchten. • Pretty-Print: Über das Menü **Source > Format** können Sie Ihr XML-Dokument mit Einrückungen und Zeilenumbrüchen lesbarer formatieren.

3.4.3 JSON

`JSON document`: Bindet JSON-Dokumente in den Prozess ein.

Eigenschaften

`Document`	Pfad zur verknüpften JSON-Datei im Repository
`Label`	Beschriftung des Symbols im Prozessdiagramm
`Operation`	Bestimmt, ob die Datei gelesen oder geschrieben werden soll **Mögliche Werte:** • *`Read`*: Datei in den Prozess einlesen und Daten an den nächsten Prozessschritt weitergeben (Standard) • *`Write`*: Datei mit den Daten aus dem vorangegangenen Prozessschritt speichern
`Type`	Bestimmt den Typ der Datei **Mögliche Werte:** • `Resource` • `Buffer` (Standard)

JSON-Dokument erstellen

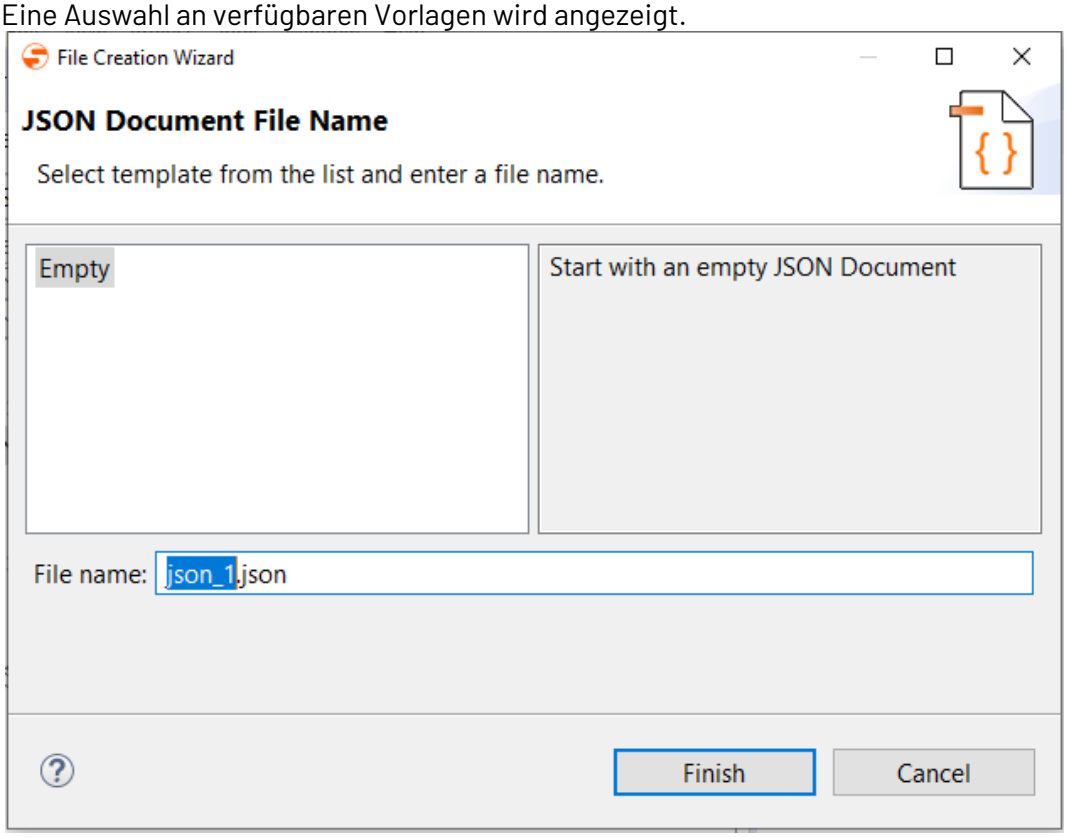
JSON-Dokumente lassen sich innerhalb eines ESB-Projekts im Ordner `Resources` (für Ressourcen) und im Ordner `TemporaryFiles` (für Puffer-Dateien) anlegen. Hierzu stehen verschiedene Möglichkeiten zur Verfügung:

- über das Menü **Project > Add JSON Document**
- über das Kontextmenü **New > JSON Document**
- über das Menü **New > Add JSON Document**

Um ein JSON-Dokument anzulegen, gehen Sie wie folgt vor:

1. Klicken Sie im ESB-Projekt mit der rechten Maustaste auf den Ordner, um das Kontextmenü aufzurufen.

2. Rufen Sie im Kontextmenü **New > JSON Document** auf.
   Eine Auswahl an verfügbaren Vorlagen wird angezeigt.



3. Wählen Sie eine leere Vorlage oder eine vorhandene Vorlage.
4. Geben Sie in **File Name** den Dateinamen ein.
5. Klicken Sie auf **Finish**, um das JSON-Dokument anzulegen.
   Das JSON-Dokument wurde angelegt und wird im JSON Editor geöffnet.

3.4.3.1 JSON Editor

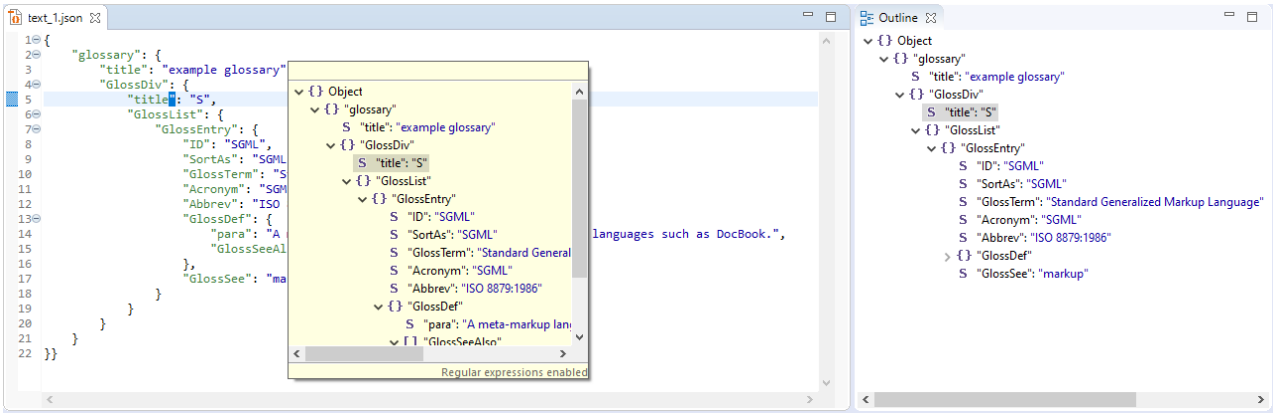
Mit dem JSON Editor können Sie JSON-Dokumente direkt im X4 Designer bearbeiten.

> ⓘ Um den JSON Editor zu starten, legen Sie entweder ein neues JSON-Dokument an oder öffnen ein bereits vorhandenes Dokument.

In der Sicht **Outline** wird die Struktur des geöffneten JSON-Dokuments angezeigt. Zudem können Sie über das Kontextmenü **Quick Outline** oder über die Tastenkombination `STRG+O` die Sicht **Quick Outline** öffnen, um innerhalb des JSON-Dokuments nach Schlüsseln und Werten zu suchen.

Tastenkombinationen

`STRG+O`	Sicht **Quick Outline** öffnen
`STRG+Umschalttaste+F`	JSON-Code formatieren



Die Formatierung des JSON-Codes sowie die Farben zur Syntaxhervorhebung können Sie unter **Tools > Options > JSON Editor** ändern. Siehe auch JSON-Editor konfigurieren.

3.4.4 Document

`Document`: Bindet beliebige Nicht-XML-Daten, wie z. B. ASCII-Text oder Binär-Formate, in den Prozess ein. Dabei wird versucht, den MIME-Typ aus der Dateinamensendung zu ermitteln. Als Zeichenkodierung wird die System-Zeichenkodierung verwendet.

Wenn Zeichenkodierung und MIME-Typ nicht korrekt ermittelt werden, dann verwenden Sie den Document Properties Changer, um diese entsprechend zu setzen.

Eigenschaften

`Document`	Pfad zur verknüpften Text- oder Binär-Datei im Repository
`Label`	Beschriftung des Symbols im Prozessdiagramm
`Operation`	Bestimmt, ob die Datei gelesen oder geschrieben werden soll **Mögliche Werte:** • *`Read`*: Datei in den Prozess einlesen und Daten an den nächsten Prozessschritt weitergeben (Standard) • *`Write`*: Datei mit den Daten aus dem vorangegangenen Prozessschritt speichern

`Type`	Bestimmt den Typ der Datei **Mögliche Werte:** • `Resource` • `Buffer` (Standard)

3.4.4.1 ASCII-Deskriptor

Ein (ASCII-)Deskriptor ist ein XML-Dokument, das die Struktur von Datensätzen in einer Textdatei beschreibt und u. a. den Adaptern ASCII Writer, COBOL Copybook Writer und COBOL Copybook Reader als Regeldatei zur Konvertierung zwischen Text- und XML-Daten dient. Der Repository-Pfad von Deskriptor-Dokumenten wird in diesen Adaptern als Parameter angegeben. Deskriptoren werden also in der Regel nicht als eigener Baustein in Prozesse eingebunden.

> Für Textdateien im *Cobol Copybook*-Format können Sie ein spezifisches Deskriptor-Dokument über den Cobol Copybook to XML Wizard automatisch erzeugen. Für CSV-Textdokumente verwenden Sie den CSV Converter.

Elemente

`<Descriptor>`	Wurzelelement. Enthält ein oder mehrere Elemente `<Record>` sowie die unten genannten Attribute.
`<Record>`	Entspricht einem Datensatz in der Textdatei. Kann ein oder mehrere Elemente `<Column>` oder `<Record>` sowie die unten genannten Attribute enthalten.
`<Column>`	Entspricht einer Spalte bzw. einem konkreten Wert in der Textdatei und besitzt die unten genannten Attribute, darf aber keine Kindelemente besitzen.

Attribute von <Descriptor>

Attribut	Beschreibung
`Id`	ID des Datensatzes (optional) > Das Attribut wird im ASCII Reader und ASCII Writer verwendet.

Attribut	Beschreibung
`Name`	Name des Wurzelelements im XML-Dokument ⓘ Das Attribut wird im GDV Message Writer sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet.
`Description`	Beschreibungstext für den ASCII-Deskriptor (optional)

`RecordSeparator`	Trennzeichen für die einzelnen Datensätze (optional)
	
	ⓘ • Das Attribut wird im GDV Message Writer, im ASCII Reader (`EscapedString`) und ASCII Writer (`EscapedString`) sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet.
	• Im Falle von `EscapedString` werden einige Escape-Zeichen automatisch ersetzt:
	  • `\n`: neue Zeile (`newline`)
	  • `\r`: Wagenrücklauf (`carriage return`)
	  • `\t`: Tab – Cursor um einen vorgegebenen Abstand weiterrücken (`tab`)
	  • `\u....`: Hexadezimalzahlen mit 4 Ziffern werden in ganze Zahlen umgewandelt, z.B. `\u00FF` zu *255*

`Columns`	Erforderliche Spaltenanzahl des Datensatzes (optional) ⓘ Das Attribut wird im ASCII Reader (`integer`) und ASCII Writer (`integer`) verwendet.
`RecordOrder`	Reihenfolge der einzelnen Datensätze (optional) **Mögliche Werte:** • `Fix`: Die Reihenfolge bleibt wie in der Textdatei. • `Loose`: Die Reihenfolge aus dem ASCII-Deskriptor wird übernommen. ⓘ Das Attribut wird im ASCII Reader und ASCII Writer verwendet.

Attribut	Beschreibung
`RecordLength`	Länge der einzelnen Datensätze (optional) ⓘ Das Attribut wird im GDV Message Reader (`integer`) verwendet.
`Identifier`	Wert des Datensatz-Identifikators (optional) ⓘ Das Attribut wird im ASCII Reader (`String`) und ASCII Writer (`String`) verwendet.
`IdentifierPos`	Position der Datensatz-Identifikatoren (optional) ⓘ Das Attribut wird im GDV Message Reader (`integer`), im ASCII Reader (`integer`) und ASCII Writer (`integer`) sowie im COBOL Copybook Writer (`integer`) und COBOL Copybook Reader (`integer`) verwendet.
`IdentifierLength`	Länge der Datensatz-Identifikatoren (optional) ⓘ Das Attribut wird im GDV Message Reader (`integer`), im ASCII Reader (`integer`) und ASCII Writer (`integer`) sowie im COBOL Copybook Writer (`integer`) und COBOL Copybook Reader (`integer`) verwendet.
`TeilsatzPos`	Gibt die Position innerhalb der Record-Daten an ⓘ Das Attribut wird im GDV Message Reader (`integer`) verwendet. **Mögliche Werte:** • `Identifier = TeilsatzPos`: Sub-Records werden gelesen. • `0`: Es werden alle Sub-Records gelesen.

Attribut	Beschreibung

`Suffix`	Suffix-Angabe (optional). Markiert das Ende eines Datensatzes.
	
	• Das Attribut wird im ASCII Reader (`EscapedString`) und ASCII Writer (`EscapedString`) verwendet.
	• Im Falle von `EscapedString` werden einige Escape-Zeichen automatisch ersetzt:
	  • `\n`: neue Zeile (`newline`)
	  • `\r`: Wagenrücklauf (`carriage return`)
	  • `\t`: Tab – Cursor um einen vorgegebenen Abstand weiterrücken (`tab`)
	  • `\u....`: Hexadezimalzahlen mit 4 Ziffern werden in ganze Zahlen umgewandelt, z.B. `\u00FF` zu *255*

`Prefix`	Präfix-Angabe (optional). Markiert den Beginn eines Datensatzes.
	
	• Das Attribut wird im ASCII Reader (`EscapedString`) und ASCII Writer (`EscapedString`) verwendet.
	• Im Falle von `EscapedString` werden einige Escape-Zeichen automatisch ersetzt:
	  • `\n`: neue Zeile (`newline`)
	  • `\r`: Wagenrücklauf (`carriage return`)
	  • `\t`: Tab – Cursor um einen vorgegebenen Abstand weiterrücken (`tab`)
	  • `\u....`: Hexadezimalzahlen mit 4 Ziffern werden in ganze Zahlen umgewandelt, z.B. `\u00FF` zu *255*

`Mandatory`	Datensatz als Pflichtdatensatz deklarieren (optional) Das Attribut wird im ASCII Reader und ASCII Writer verwendet. **Mögliche Werte:** • `true`: Pflichtdatensatz • `false`: kein Pflichtdatensatz

Attribut	Beschreibung
`Multiple`	Datensatz kann mehrfach vorhanden sein (optional) ⓘ Das Attribut wird im ASCII Reader und ASCII Writer verwendet. **Mögliche Werte:** • `true`: Datensatz kann mehrfach vorhanden sein (siehe Attribut `Occurs`). Das Attribut `Occurs` muss hierbei einen Wert enthalten, da sonst der Fehler *`java.lang.NumberFormatException: null`* ausgegeben wird. • `false`: Datensatz darf nur einmal vorhanden sein.
`Length`	Erforderliche Länge des Datensatzes in Zeichen (optional) ⓘ Das Attribut wird im ASCII Reader (`LengthFormat`) und ASCII Writer (`LengthFormat`) verwendet.

`Separator`	Trennzeichen für die einzelnen Datensätze (optional)
	
	ⓘ • Das Attribut wird im ASCII Reader (`EscapedString`) und ASCII Writer (`EscapedString`) verwendet.
	• Im Falle von `EscapedString` werden einige Escape-Zeichen automatisch ersetzt:
	  • `\n`: neue Zeile (`newline`)
	  • `\r`: Wagenrücklauf (`carriage return`)
	  • `\t`: Tab – Cursor um einen vorgegebenen Abstand weiterrücken (`tab`)
	  • `\u....`: Hexadezimalzahlen mit 4 Ziffern werden in ganze Zahlen umgewandelt, z.B. `\u00FF` zu *255*

Attribute von <Record>

Attribut	Beschreibung
`Id`	ID des Datensatzes (optional) ⓘ Das Attribut wird im ASCII Reader (`String`) und ASCII Writer (`String`) sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet.

Attribut	Beschreibung
`Name`	Name der Datensatz-Elemente im XML-Dokument Das Attribut wird im GDV Message Reader und im GDV Message Writer, im ASCII Reader (`String`) und ASCII Writer (`String`) sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet.

`Prefix`	Präfix-Angabe (optional). Markiert den Beginn eines Datensatzes.
	
	• Das Attribut wird im ASCII Reader (`EscapedString`) und ASCII Writer (`EscapedString`) sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet.
	• Im Falle von `EscapedString` werden einige Escape-Zeichen automatisch ersetzt:
	  • `\n`: neue Zeile (`newline`)
	  • `\r`: Wagenrücklauf (`carriage return`)
	  • `\t`: Tab – Cursor um einen vorgegebenen Abstand weiterrücken (`tab`)
	  • `\u....`: Hexadezimalzahlen mit 4 Ziffern werden in ganze Zahlen umgewandelt, z.B. `\u00FF` zu *255*

`Suffix`	Suffix-Angabe (optional). Markiert das Ende eines Datensatzes.
	
	• Das Attribut wird im ASCII Reader (`EscapedString`) und ASCII Writer (`EscapedString`) sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet.
	• Im Falle von `EscapedString` werden einige Escape-Zeichen automatisch ersetzt:
	  • `\n`: neue Zeile (`newline`)
	  • `\r`: Wagenrücklauf (`carriage return`)
	  • `\t`: Tab – Cursor um einen vorgegebenen Abstand weiterrücken (`tab`)
	  • `\u....`: Hexadezimalzahlen mit 4 Ziffern werden in ganze Zahlen umgewandelt, z.B. `\u00FF` zu *255*

Attribut	Beschreibung
`Mandatory`	Datensatz als Pflichtdatensatz deklarieren (optional) ⓘ Das Attribut wird im ASCII Reader und ASCII Writer sowie im COBOL Copybook Writer und COBOL Copybook Reader verwendet. **Mögliche Werte:** • *`true`*: Pflichtdatensatz • *`false`*: kein Pflichtdatensatz
`Multiple`	Datensatz kann mehrfach vorhanden sein (optional) ⓘ Das Attribut wird im ASCII Reader und ASCII Writer sowie im COBOL Copybook Writer und COBOL Copybook Reader verwendet. **Mögliche Werte:** • *`true`*: Datensatz kann mehrfach vorhanden sein (siehe Attribut `Occurs`). Das Attribut `Occurs` muss hierbei einen Wert enthalten, da sonst der Fehler *`java.lang.NumberFormatException: null`* ausgegeben wird. • *`false`*: Datensatz darf nur einmal vorhanden sein.
`IdentifierPos`	Position der Datensatz-Identifikatoren (optional) ⓘ Das Attribut wird im ASCII Reader (`integer`) und im ASCII Writer (`integer`) verwendet.
`IdentifierLength`	Länge der Datensatz-Identifikatoren (optional) ⓘ Das Attribut wird im ASCII Reader (`integer`) und im ASCII Writer (`integer`) verwendet.
`Identifier`	Wert des Datensatz-Identifikators (optional) ⓘ Das Attribut wird im GDV Message Reader und im GDV Message Writer, im ASCII Reader (`String`) und ASCII Writer (`String`) sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet.

Attribut	Beschreibung

`ColumnSeparator`	Spalten-Trennzeichen (optional)
	
	• Das Attribut wird im ASCII Reader (`EscapedString`) und ASCII Writer (`EscapedString`) sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet.
	• Im Falle von `EscapedString` werden einige Escape-Zeichen automatisch ersetzt:
	  • `\n`: neue Zeile (`newline`)
	  • `\r`: Wagenrücklauf (`carriage return`)
	  • `\t`: Tab – Cursor um einen vorgegebenen Abstand weiterrücken (`tab`)
	  • `\u....`: Hexadezimalzahlen mit 4 Ziffern werden in ganze Zahlen umgewandelt, z.B. `\u00FF` zu *255*

`Columns`	Erforderliche Spaltenanzahl des Datensatzes (optional)
`ColumnOrder`	Reihenfolge der Spalten Das Attribut wird im ASCII Reader und im ASCII Writer verwendet. **Mögliche Werte:** • `Fix`: Die Reihenfolge bleibt wie in der Text-Datei. • `Loose`: Die Reihenfolge aus dem ASCII-Deskriptor wird übernommen.
`Occurs`	Anzahl der Wiederholungen des Datensatzes Das Attribut wird im ASCII Reader (`integer`) und ASCII Writer (`integer`) sowie im COBOL Copybook Writer (`integer`) und COBOL Copybook Reader (`integer`) verwendet.
`Description`	Beschreibungstext für den Datensatz (optional)
`Length`	Erforderliche Länge des Datensatzes in Zeichen (optional) Das Attribut wird im GDV Message Reader (`integer`), im ASCII Reader (`LengthFormat`) und ASCII Writer (`LengthFormat`) sowie im COBOL Copybook Writer (`integer`) und COBOL Copybook Reader (`integer`) verwendet.

Attribute von <Column>

Attribut	Beschreibung
`Id`	Spalten-ID (optional) ⓘ Das Attribut wird im ASCII Reader (`String`) und ASCII Writer (`String`) sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet.
`Name`	Name des Feld-Elements im XML-Dokument ⓘ Das Attribut wird im GDV Message Reader (`String`), im ASCII Reader (`String`) und ASCII Writer (`String`) sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet.
`Type`	Datentyp der Spalte bzw. des Felds ⓘ Das Attribut wird im GDV Message Writer (`Integer` | `String`), im ASCII Reader (`Integer` | `String`) und ASCII Writer (`Integer` | `String`) sowie im COBOL Copybook Writer (`Integer` | `String`) und COBOL Copybook Reader (`Integer` | `String`) verwendet. **Mögliche Werte:** • *`String`*: Zeichenkette • *`Number`*: Zahlenwert

`Prefix`	Präfix-Angabe (optional). Markiert den Beginn eines Feldwerts.
	
	ⓘ • Das Attribut wird im ASCII Reader (`EscapedString`) und ASCII Writer (`EscapedString`) sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet.
	• Im Falle von `EscapedString` werden einige Escape-Zeichen automatisch ersetzt:
	  • `\n`: neue Zeile (`newline`)
	  • `\r`: Wagenrücklauf (`carriage return`)
	  • `\t`: Tab – Cursor um einen vorgegebenen Abstand weiterrücken (`tab`)
	  • `\u....`: Hexadezimalzahlen mit 4 Ziffern werden in ganze Zahlen umgewandelt, z.B. `\u00FF` zu *255*

Attribut	Beschreibung
`Suffix`	Suffix-Angabe (optional). Markiert das Ende eines Feldwerts.
`Mandatory`	Feld als Pflichtfeld deklarieren (optional) ⓘ Das Attribut wird im ASCII Reader und ASCII Writer sowie im COBOL Copybook Writer und COBOL Copybook Reader verwendet. **Mögliche Werte:** • *`true`*: Pflichtfeld • *`false`*: kein Pflichtfeld
`Multiple`	Feld kann mehrfach vorhanden sein (optional) ⓘ • Das Attribut wird im ASCII Reader und ASCII Writer verwendet. **Mögliche Werte:** • *`true`*: Feld kann mehrfach vorhanden sein. • *`false`*: Feld darf nur einmal vorhanden sein.
`IdentifierPos`	Position der Spalten-Identifikatoren (optional) ⓘ Das Attribut wird im ASCII Reader (`integer`) und ASCII Writer (`integer`) verwendet.
`IdentifierLength`	Länge der Spalten-Identifikatoren (optional) ⓘ Das Attribut wird im ASCII Reader (`integer`) und ASCII Writer (`integer`) verwendet.
`Identifier`	Wert des Spalten-Identifikators (optional) ⓘ Das Attribut wird im ASCII Reader (`String`) und ASCII Writer (`String`) verwendet.

Attribut	Beschreibung
`Default`	Standardwert des Felds (nur bei Pflichtfeldern) ⓘ Das Attribut wird im ASCII Reader (`String`) und ASCII Writer (`String`) sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet.
`Regex`	Regulärer Ausdruck, der die Spaltenwerte beschreibt (optional) ⓘ Das Attribut wird im ASCII Reader (`String`) und ASCII Writer (`String`) sowie im COBOL Copybook Writer (`String`) und COBOL Copybook Reader (`String`) verwendet. *Beispiel*: *`*[0-9a-fA-F]+.*`*

`Length`	Erforderliche Feldlänge in Zeichen (optional)
	
	ⓘ • Das Attribut wird im GDV Message Reader (`integer`) und im GDV Message Writer (`integer`), im ASCII Reader (`LengthFormat`) und ASCII Writer (`LengthFormat`) sowie im COBOL Copybook Writer (`LengthFormat`) und COBOL Copybook Reader (`LengthFormat`) verwendet.
	• Im COBOL Copybook Writer und COBOL Copybook Reader sind folgende Wert-Angaben möglich:
	  • `ganze Zahlen`: z.B. 10 wird als 10,0 interpretiert
	  • `keine Angabe`: wird als 0,0 interpretiert
	  • `Dezimalzahlen`: z.B. 10.2 wird als 10,2 interpretiert

`RecordMultiplicator`	Datensatz wiederholen ⓘ Das Attribut wird im COBOL Copybook Writer und COBOL Copybook Reader verwendet. **Mögliche Werte:** • *`true`*: spezielles Feld (angegeben in `RecordMultiplicated`) wiederholen • *`false`*: Feld nicht wiederholen

Attribut	Beschreibung
`RecordMultiplicated`	ID des mehrfach auftretenden Datensatzes (falls `RecordMultiplicator` auf *true* gesetzt wurde) Das Attribut wird im COBOL Copybook Writer (`integer`) und COBOL Copybook Reader (`integer`) verwendet.
`Binary`	Beim Lesen/Schreiben werden die Daten von Base64 nach binär bzw. von binär nach Base64 kodiert Das Attribut wird im COBOL Copybook Writer und COBOL Copybook Reader verwendet.
`Occurs`	Anzahl der Wiederholungen der Spalte Das Attribut wird im COBOL Copybook Writer (`integer`) und COBOL Copybook Reader (`integer`) verwendet.

3.4.5 Subprocess

`Subprocess`: Bietet die Möglichkeit, Teile von Prozessen bzw. Teilprozesse in separate Prozessdateien auszulagern. Sie können jeden Prozess in einen anderen Prozess als Subprozess einbinden. Dabei kann ein Subprozess einen Input aus dem vorangegangenen Prozessbaustein erhalten und gibt den letzten Status innerhalb des Subprozesses sowie ggf. ein Output-Dokument an den darauf folgenden Prozessbaustein im übergeordneten Prozess weiter.

- Um den Subprozess zu bearbeiten, auf das entsprechende Symbol auf der Zeichnungsfläche doppelklicken.
- Falls ein Input- bzw. Output-Schema beim Anlegen des Prozesses hinterlegt wurde, kann dieses über den Kontextmenü-Eintrag **Show input schema** bzw. **Show output schema** aufgerufen werden.

Eigenschaften

`Document`	Relativer Pfad zum Subprozess Dieser Pfad ist über den Parameter `workflow` dynamisch parametrierbar. Die Verwendung von Platzhaltern (z. B. `%START_PROCESS_PATH%` oder `%CALLING_PROCESS_PATH%`) ist ebenfalls möglich (nicht im Debug-Modus des X4 Designers)

`Label`	Beschriftung des Symbols im Prozessdiagramm

Teilprozesse als Subprozess auslagern

Bei umfangreicheren Prozessen ist es sinnvoll, zusammenhängende Bausteine als Subprozesse auszulagern. So wird der Prozess übersichtlicher, und die Subprozesse lassen sich mehrfach verwenden. Prozesse lassen sich praktisch beliebig tief ineinander verschachteln. Subprozesse werden als separate `.wrf`-Dateien gespeichert.

> Prozesse lassen sich wie andere Prozessbausteine per Drag&Drop aus der Sicht **Projects** in die Zeichnungsfläche des Process Editors ziehen.

1. Markieren Sie mindestens zwei Bausteine inklusive aller Übergangslinien, die ausgelagert werden sollen.
2. Rufen Sie die Funktion **Pack As Subprocess** im Kontextmenü auf:
   - Geben Sie in **File name** den Namen des Subprozesses ein.
   - Klicken Sie auf **Finish**.
3. Der Subprozess mit den markierten Bausteinen wird als eigenständige Prozessdatei in der Sicht **Projects** angelegt. Im Hauptprozess werden die gewählten Bausteine automatisch durch den neu angelegten Subprozess-Baustein ersetzt.

3.4.5.1 Private Subprozesse verwenden

Subprozesse können privat verwendet werden. Private Subprozesse können nur für ihren Elternprozess wiederverwendet werden. Ein privater Subprozess kann entweder neu erstellt oder aus bestehenden Prozessbausteinen gepackt werden.

Um einen **privaten Subprozess zu erstellen**, gehen Sie wie folgt vor:

1. Wählen Sie in der Sicht **Projects** einen bestehenden Prozess aus.
2. Klicken Sie im Menü auf **Project** > **Add Process...**.

   > **Alternativ**
   > Klicken Sie mit der rechten Maustaste auf einen bestehenden Prozess. Klicken Sie im Kontextmenü auf **New** > **Process**.

   Ein neuer Prozess wird als privater Subprozess des bestehenden Prozesses angelegt und kann bearbeitet werden.

Um einen **privaten Subprozess** aus bestehenden Prozessbausteinen zu **packen**, gehen Sie wie folgt vor:

1. Öffnen Sie einen bestehenden Prozess.
2. Markieren Sie mindestens zwei Bausteine inklusive aller Übergangslinien.
3. Klicken Sie mit der rechten Maustaste auf die markierten Bausteine.

4. Wählen Sie im Kontextmenü **Pack as Private Subprocess**.
5. Geben Sie in **File name** den Namen des Subprozesses ein.
6. Bestätigen Sie mit **Finish**.
   Die markierten Bausteine werden als privater Subprozess angelegt und können dort bearbeitet werden.

3.4.6 Mapping

`Mapping`: Transformiert ein XML-Dokument in einer Quellstruktur über ein XSLT-Stylesheet in eine Zielstruktur (meist XML) und ermöglicht so, die einzelnen Datenelemente einander zuzuordnen (sog. *Mapping*). XSLT dient der X4 BPMS zudem als Skriptsprache, mit der Prozessbausteine dynamisch parametriert werden können.

Eigenschaften

`Document`	Pfad zur verknüpften Datei im Repository
`Label`	Beschriftung des Symbols im Prozessdiagramm
`Component Parameters`	Component Parameter setzen. Weitere Informationen hierzu finden Sie unter Component Parameter verwenden.
`Empty Input` *(X4EMPTYINPUT)*	Keine Daten aus dem vorangehenden Prozessschritt im XSL-Mapping verarbeiten **Mögliche Werte:** • *`true`*: keine Input-Daten verarbeiten ⓘ Wenn `Empty Input` auf `true` gesetzt ist, wird der `Mapping`-Baustein zusätzlich gekennzeichnet. • *`false`*: Input-Daten verarbeiten (Standard)
`Skip Dynamic Params` *(X4SKIPDYNPARAM)*	Auswertung von dynamischen Parametern abschalten. Dies kann die Ausführung von XSL-Mappings deutlich beschleunigen. **Mögliche Werte:** • *`true`*: Auswertung von dynamischen Parameter abschalten ⓘ Wenn `Skip Dynamic Params` auf `true` gesetzt ist, wird der `Mapping`-Baustein zusätzlich gekennzeichnet. • *`false`*: Dynamische Parameter im XSL-Mapping verarbeiten (Standard)

`Remove X4 Processing Instructions`	X4 Processing Instructions im Mapping-Ergebnis entfernen **Mögliche Werte:** • *`true`*: X4 Processing Instructions aus dem Ergebnis löschen (Standard) • *`false`*: X4 Processing Instructions im Ergebnis anzeigen ⓘ Wenn `Remove X4 Processing Instructions` auf `false` gesetzt ist, wird der `Mapping`-Baustein zusätzlich gekennzeichnet.

3.4.6.1 XSL-Mapping anlegen und verwenden

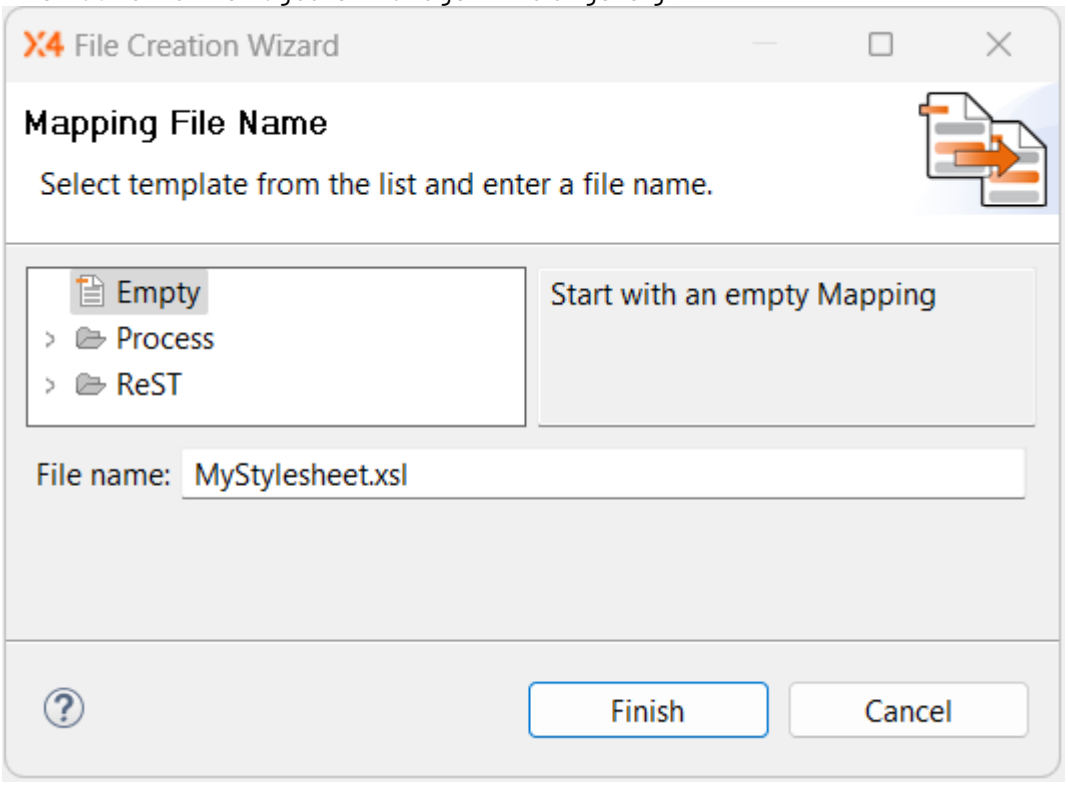
Mappings lassen sich innerhalb eines ESB-Projekts im Ordner **Transformations** anlegen.

Hierzu stehen zwei Möglichkeiten zur Verfügung:

- über das Menü **Project** > **Add XSL Stylesheet**
- über das Kontextmenü **New** > **XSL Stylesheet**

So legen Sie ein XSL-Mapping an:

1. Klicken Sie im ESB-Projekt mit der rechten Maustaste auf den Ordner **Transformations,** um das Kontextmenü aufzurufen.
2. Rufen Sie im Kontextmenü **New** > **XSL Stylesheet** auf.
   Eine Auswahl an verfügbaren Vorlagen wird angezeigt.



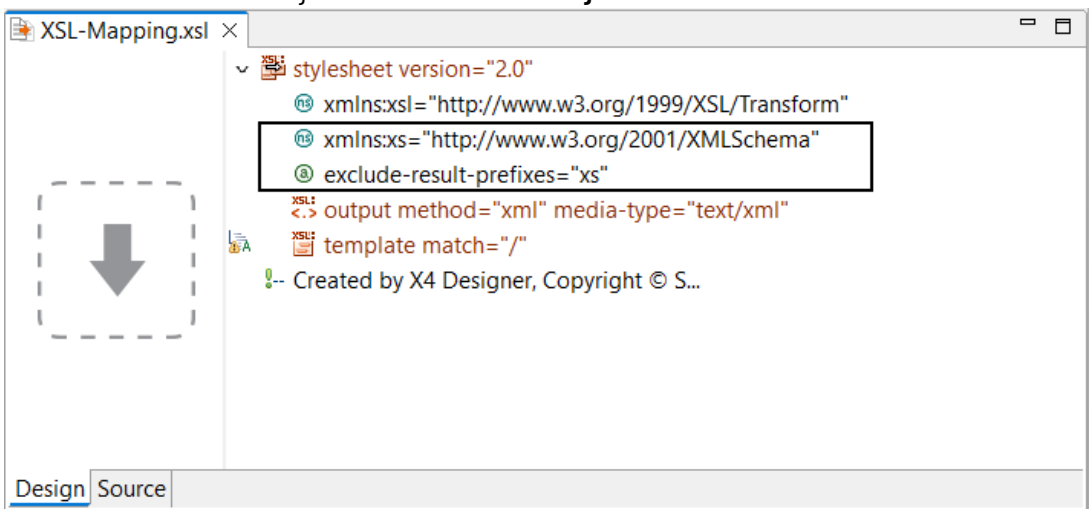
3. Wählen Sie eine leere Vorlage oder eine vorhandene Vorlage.
4. Geben Sie in **File Name** den Dateinamen ein.
5. Wählen Sie **Finish** aus, um das XSL-Mapping anzulegen.
   Das XSL-Mapping wurde angelegt und wird im Mapping Editor geöffnet.

3.4.6.1.1 Quell- und Zieldatenstruktur im XSL-Mapping definieren

Im ersten Schritt legen Sie die Quelldatenstruktur der Transformation fest. Das heißt, dass Sie den linken Bereich des XSL-Mappings mit Quelldaten befüllen.

Dazu haben Sie zwei Möglichkeiten:

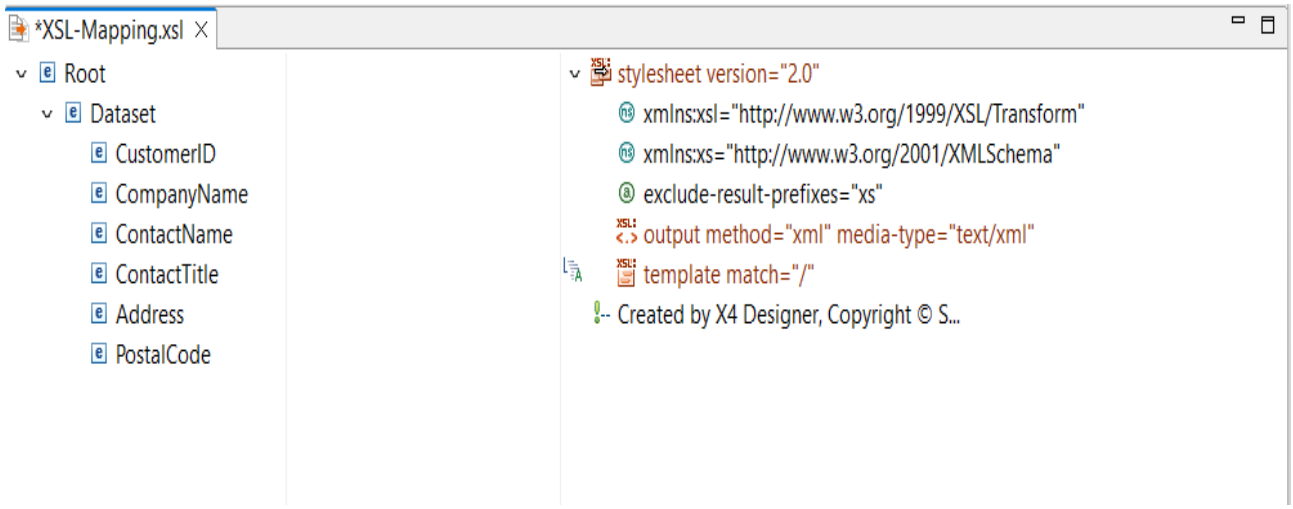
1. Ziehen Sie ein XML-Quelldokument (XML-Dokument oder XML-Schema) aus der Sicht **Projects** per Drag&Drop in den Quelldokument-Bereich des Mapping Editors.
   Alternativ können Sie mit der rechten Maustaste in den Quelldokument-Bereich des XSL-Mappings klicken und **Assign Input** auswählen, um eine Datei zum Einfügen auszuwählen. Zur Auswahl stehen alle Projekte aus der Sicht **Projects.**



> Der Namespace `xmlns:xs="http://www.w3.org/2001/XMLSchema"` und das Attribut `exclude-result-prefixes="xs"` ermöglichen die Verwendung von Datentypen in Verbindung mit XPath 2.0. Beide Einträge sind nur in neu angelegten XSL-Stylesheets enthalten, die nicht auf einer Vorlage basieren.

2. Klicken Sie mit der rechten Maustaste in den Quelldokument-Bereich des XSL-Mappings, und wählen Sie **Paste Input**, um den Inhalt der Zwischenablage einzufügen.
   Beispiel: Sie können beim Debuggen eines Prozesses die Daten aus der Sicht **Output** in die Zwischenablage kopieren und hier einfügen, oder Sie können XML-Daten aus einem Editor kopieren und hier einfügen.

Das eingefügte XML-Quelldokument wird in einer Baumansicht dargestellt:



Im nächsten Schritt legen Sie die Grundstruktur des Zieldokuments im XSL-Mapping an. Dazu gehen Sie wie folgt vor:

1. Klicken Sie im Stylesheet-Bereich mit der rechten Maustaste auf das Root-Template (`template match="/"`), um das Kontextmenü zu öffnen.
2. Wählen Sie im Kontextmenü **XSL Element**, um den **XSL Element Creation Wizard** zu öffnen.
3. Fügen Sie alle benötigten Elemente mit der entsprechenden Hierarchie ein.
   Beispiel: Die Struktur des Zieldokuments ist ein HTML-Grundgerüst (`<html>`, `<body>`), das eine Tabelle (`<table>`) mit einer Zeile (`<tr>`) mit drei Kopfzellen (`<th>`) und einer weiteren Zeile (`<tr>`) mit drei Zellen (`<td>`) enthält.

   > Wenn Sie bereits die XML-Struktur kennen, die durch das XSL-Mapping erzeugt werden soll, dann können Sie per Drag&Drop aus der Sicht **Projects** ein XML-Dokument oder ein XML-Schema zuweisen. Dieses können Sie dann als Zielstruktur für das Mapping verwenden.
   > Beim Zuweisen einer XML-Struktur per Drag&Drop werden Sie gefragt, wie diese eingefügt werden soll:
   > - **Full XML structure including data** fügt die komplette XML-Dokumentstruktur inklusive Element- und Attributwerte ein.
   > - **Virtual nodes (structure without data)** fügt lediglich die Baumstruktur als virtuelle Knoten in den Stylesheet-Bereich ein. Erst per Doppelklick oder über ein Daten-Mapping auf einen virtuellen Knoten werden diese tatsächlich in der Zielstruktur erzeugt.

4. Wenn Sie Text hinzufügen möchten: Wählen Sie im Kontextmenü **Add Other Nodes > Text**.
   Beispiel: Kopfzeilen-Beschriftung für die Tabellenkopfzellen einfügen.

3.4.6.1.2 Daten-Mapping: *for-each* verwenden

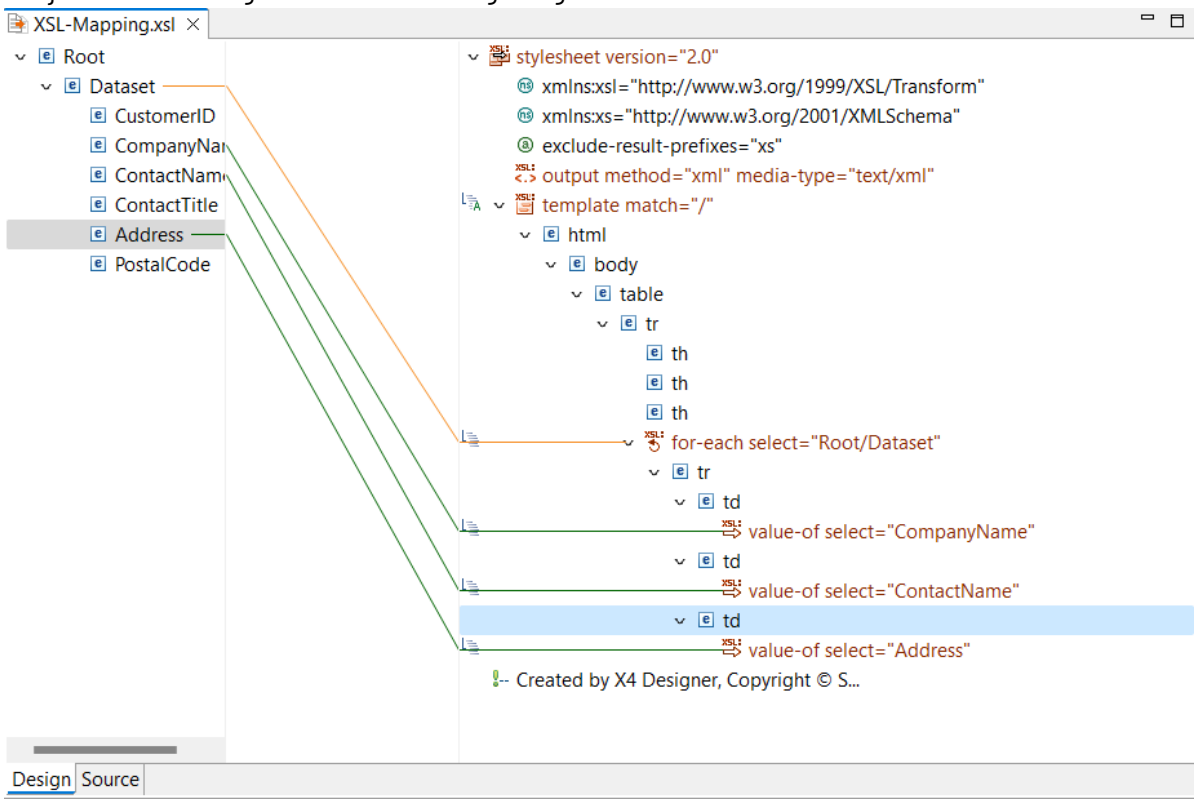
So bauen Sie mit `<xsl:for-each>` eine Zielstruktur auf, die sich pro Datensatz wiederholt:

1. Markieren Sie im Quelldokument die wiederholt vorkommende Knotenstruktur. Halten Sie die `Shift`-Taste gedrückt, und ziehen Sie die Struktur per Drag&Drop auf das zu wiederholende Zielelement.
   Ein `for-each`-Element wird erzeugt.
   Beispiel: Verknüpfen Sie `<Dataset>` auf diese Weise mit dem zweiten `<tr>`-Element, um HTML-Tabellenzeilen dynamisch zu erzeugen.
2. Stellen Sie sicher, dass das `for-each`-Element unmittelbarer Elternknoten des zweiten `<tr>`-Elements ist.

3.4.6.1.3 Daten-Mapping: *value-of* verwenden

So übernehmen Sie mit `<xsl:value-of>` Knoteninhalte vom Quelldokument in das Zieldokument:

1. Klicken Sie den Quellknoten an, und ziehen Sie ihn per Drag&Drop auf den Zielknoten.
   Beispiel: Weisen Sie das`<CustomerName>`-Element dem ersten `<td>`-Element, das Element `<CompanyName>` dem zweiten `<td>`-Element und das Element `<Address>` dem dritten `<td>`-Element zu.
2. Ziehen Sie weitere Verbindungslinien.
   Für jede Verbindung wird eine Linie angezeigt.



3.4.6.1.4 Daten-Mapping: *copy-of* verwenden

So kopieren Sie mit `<xsl:copy-of>` einen Knoten inklusive aller Kindknoten (Elemente, Attribute, Namespace-Definitionen etc.):

1. Markieren Sie im Quelldokument den zu kopierenden Knoten.
2. Halten Sie die `Strg`-Taste gedrückt, und ziehen Sie den Knoten per Drag&Drop auf das Zielelement.
   Ein `copy-of`-Element wird erzeugt.

3.4.6.1.5 Source-Ansicht: XSLT-Funktionen verwenden

> **Beispiel**
> Die Elementinhalte von `ContactName` und `ContactTitle` sollen über die Funktion `concat` zusammengeführt werden.

1. Wählen Sie die Registerkarte **Source**, um das XSL-Mapping im Quelltext anzuzeigen.
2. Bearbeiten Sie die XSLT-Syntax.
   Beispiel: Ersetzen Sie die Zeile `<xsl:value-of select="ContactName"></xsl:value-of>` durch `<xsl:value-of select="concat(ContactName, ', ', ContactTitle)"/>`.

3.4.6.1.6 Ausgabeformat ändern

1. Passen Sie in Element `<xsl:output>` in Attribut `method="xml"` das Ausgabeformat an.
   Beispiel: Zum Erzeugen eines HTML-Dokuments `method="html"` setzen.
2. Passen Sie in Element `<xsl:output>` in Attribut `media-type="text/xml"` den MIME-Typ des Ausgabedokuments an.
   Beispiel: Zum Erzeugen eines HTML-Dokuments `mime-type="text/html"` setzen.

3.4.6.1.7 XSL-Mapping testen

Wenn ein Input-Dokument verfügbar ist, können Sie die integrierte Vorschaufunktion des Mapping Editors verwenden.

1. Speichern Sie das XSL-Mapping mit `Strg+S`.
2. Klicken Sie in der Symbolleiste auf ▶ und anschließend auf **Run Transformation**, um das XSL-Mapping lokal auszuführen.
   Das Transformationsergebnis wird nun in der Sicht **Output** angezeigt.

3.4.6.1.8 XSL-Mapping in einen Prozess einbauen

1. Ziehen Sie das XSL-Mapping aus der Sicht **Projects** per Drag&Drop in den geöffneten Prozess.
   Beispiel: Zwischen XML-Dokument und `Stop`-Baustein einfügen.

2. Definieren Sie bei Bedarf Parameter in der Eigenschaft `Component Parameters`.
   Diese werden beim Ausführen des Prozesses an XSL-Mapping übergeben. Weitere Informationen hierzu finden Sie unter Prozessbausteine parametrieren.

3.4.6.2 Mapping Editor

Mit dem Mapping Editor können Sie XSL-Mappings in einer grafischen Ansicht erstellen und bearbeiten. Eine XML-Datenstruktur kann hier mithilfe von XSLT in eine andere XML-Datenstruktur oder in ein XSL-FO-Dokument transformiert werden. Der Mapping Editor unterstützt Sie beim Erstellen eines XSL-Mappings durch eine grafische Oberfläche und durch eine automatische Gültigkeitsprüfung des Stylesheets sowie der XPath-Ausdrücke.

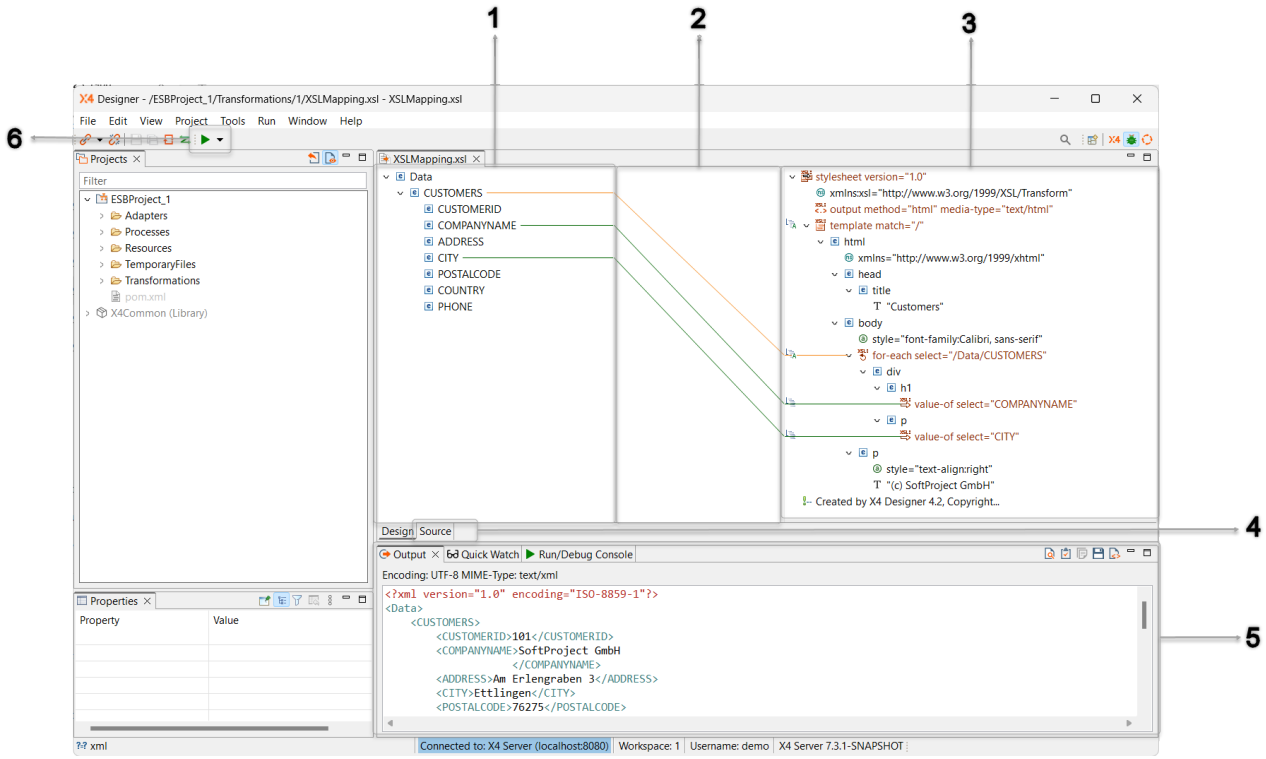
Bei einem XSL-Mapping sind drei Dokumente involviert:

- XML-Quelldokument (oder ein XML-Schema mit der Strukturinformation)
- XSLT-Stylesheet (dieses wird im Mapping Editor erstellt)
- XML-Ausgabedokument

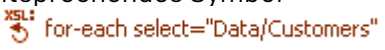
> Um den Mapping Editor zu starten, legen Sie entweder ein neues Mapping an oder öffnen ein bereits vorhandenes XSLT-Mapping.

Aufbau des Mapping Editors

Der Mapping Editor ist in folgende Bereiche unterteilt:



1	**Quelldokument-Bereich**	In diesem Bereich wird das zu transformierende Quelldokument oder eine XML-Schema-Datei (`.xsd`) angezeigt. Über das Kontextmenü können Sie den gewählten Knoten wie folgt bearbeiten: • **Copy XPath for this element to Clipboard**: XPath-Ausdruck zum entsprechenden Element in die Zwischenablage kopieren und an anderer Stelle verwenden • **Assign Input**: Input-XML-Dokument zuweisen • **Remove Input**: Input-XML-Dokument entfernen • **Paste Input**: Input-XML-Dokument aus der Zwischenablage einfügen
2	**Verknüpfungsbereich**	In diesem Bereich werden die Verknüpfungen zwischen Knoten im Quelldokument und XSL-Stylesheet als verschiedenfarbige Linien visualisiert. ⓘ Unter Tastenkombinationen im Mapping Editor finden Sie nützliche Tastenkombinationen für die Arbeit mit XSL-Stylesheets in der grafischen Ansicht des X4 Mapping Editors.

3	**Stylesheet-Bereich**	In diesem Bereich bearbeiten Sie den Hauptbestandteil eines Mappings, das XSL-Stylesheet. Das Stylesheet wird in einer Baumstruktur angezeigt und bietet über das Kontextmenü Zusatzfunktionen für die XSL-Bearbeitung.
		
		Die verschiedenen Bestandteile des XSL-Stylesheets werden in der Baumansicht mit Symbolen und farblichen Hervorhebungen dargestellt:
		
		• Knoten mit XSL-Anweisungen sind durch eine braune Schriftfarbe und ein entsprechendes Symbol gekennzeichnet, z. B.  .
		• Knoten mit einem XPath-Ausdruck sind durch ein entsprechendes XPath-Symbol ( ) gekennzeichnet. Um den XPath-Ausdruck zu bearbeiten, klicken Sie auf dieses Symbol.
		• Virtuelle Knoten (aus der Quellstruktur übernommene, aber noch nicht im Stylesheet explizit vorhandene Knoten) werden grau dargestellt.
		
		Über das Kontextmenü können Sie im Stylesheet-Bereich den gewählten Knoten wie folgt bearbeiten:
		
		• **Edit XSL Element**: XSL-Element bearbeiten
		• **Rename**: Knoten umbenennen
		• **Element**: Element einfügen
		• **Attribute**: Attribut einfügen
		• **Edit**: (bei Processing Instructions, Namespace-Deklarationen): Knoteninhalt bearbeiten
		• **XSL Element**: XSL-Element einfügen
		  • `as Child`: als Kind-Knoten
		  • `Before`: als Knoten vor dem markierten Knoten
		  • `After`: als Knoten hinter dem markierten Knoten
		• **Add other nodes**: andere Knoten einfügen (`Namespace`, `Text`, `Comment`, `Processing Instruction`)
		• **Add here**: (nur bei virtuellen Knoten): virtuellen Knoten in tatsächlichen Knoten umwandeln
		• **Add Processing Instructions**: Processing Instructions hinzufügen, um Parameter dynamisch zu setzen
		  • `Condition State` für Statusinformationen
		  • `Log Info` für Log-Nachrichten
		  • `Dynamic Parameter` zum Setzen von dynamischen Werten
		  • `No Result` für leere Mapping-Ausgaben
		• **Add Parameter**: System-, Projekt- oder globale Parameter einfügen, siehe Parameter erstellen und verwenden
		• **Add Parent**: Eltern-Element (übergeordnetes Element) einfügen
		• **Cut/Copy/Paste**: Auswahl ausschneiden, kopieren und einfügen

		• **Delete:** Knoten löschen • **Expand/Collapse**: Unterelemente des Knotens aufklappen bzw. zuklappen • **Assign Virtual Nodes**: virtuellen Knoten aus einem XML-Dokument einfügen • **Remove Virtual Node**: eingefügten virtuellen Knoten entfernen Mit dem **XSL Element Creation Wizard** können Sie einfach und schnell XSL-Mappings erstellen. Der Wizard bietet neben der kompletten Palette an XSLT 2.0-Elementen auch die Möglichkeit, die Mappings bequem über das Kontextmenü anzulegen. Über den Wizard lässt sich festlegen, ob das neue Element vor oder nach dem ausgewählten Element oder als Kind des ausgewählten Elements angelegt werden soll. Funktionen wie Autovervollständigung oder die Anzeige aller verfügbaren und erlaubten XSL-Elemente ermöglichen schnelleres Arbeiten. Zudem können die Attribute des XSL-Elements gesetzt bzw. nach dem Anlegen bearbeitet werden. Zwingend erforderliche Attribute werden dabei automatisch übernommen, optionale Attribute können je nach Bedarf ausgewählt und befüllt werden. Bei Attributen, die mehrere Optionen zur Auswahl haben, wird eine Liste mit den verfügbaren Werten bereitgestellt.
4	**Source- und Design-Ansicht**	Der Mapping Editor verfügt über zwei Ansichten, zwischen denen Sie über zwei Registerkarten wechseln können: • **Design**: In der dreispaltigen Design-Ansicht wird das XSLT-Stylesheet als Baum dargestellt. Hier können Sie die Knoten des Quelldokuments den Knoten im XSLT-Stylesheet grafisch zuordnen. • **Source**: In dieser Quellcode-Ansicht können Sie das XSLT-Stylesheet direkt im XML-Editor bearbeiten.
5	**Output**	Das Ergebnis von lokal ausgeführten XSL-Mappings wird in der Sicht **Output** angezeigt. Außerdem können Sie über die Symbolleiste die folgenden Aktionen ausführen: • *Save content of output view*: Daten aus der Sicht **Output** speichern • *Format the content of the output view*: Daten aus der Sicht **Output** formatieren

6	**Run Transformation**	Über ▶ **Run Transformation** wird das XSL-Mapping lokal ausgeführt, und das Transformationsergebnis wird in der Sicht **Output** (5) angezeigt.

3.4.7 PDF Creator

`PDF Creator`: Transformiert ein XML-Dokument mithilfe eines XSLT-Stylesheets in ein XSL-FO-Dokument. Dieses wird im selben Prozessschritt über den XSL-FO-Prozessor *Apache FOP 1.0* in ein seitenorientiertes Ausgabemedium umgewandelt. Mögliche Ausgabeformate für Reports sind PDF, PDF/A und PostScript.

Eigenschaften

`Document`	Pfad zur verknüpften Datei im Repository
`Component Parameters`	Component Parameter. Weitere Informationen hierzu finden Sie unter Component Parameter verwenden.
`Report type`	Ausgabeformat des Reports **Mögliche Werte:** • *PDF*: PDF-Dokument (Portable Document Format) • *PDF/A*: PDF-Dokument im Archivierungsformat *PDF/A-1b* gemäß *ISO 19005-1*; siehe Hinweis unten • *PS*: PostScript-Datei ⓘ Für die PDF/A-Option fordert Apache FOP 1.0 für jeden Textblock innerhalb des FO-Dokuments explizit die Festlegung einer Schriftart über das Attribut `font-family`. Zudem müssen sämtliche verwendeten Schriftarten auf dem System während der PDF/A-Erzeugung verfügbar sein.

Reports erstellen mit XSL-FO

Mit dem Baustein `PDF Creator` können Sie zur Laufzeit eines Prozesses dynamisch Dokumente für die Druckausgabe (Reports) generieren. Ein Report ist ein spezielles XSL-Mapping, das XML-Daten transformiert und ein XML-Dokument mit XSL-FO-Anweisungen ausgibt. Das erzeugte XSL-FO-Dokument wird der X4 BPMS direkt in das angegebene Ausgabeformat umgewandelt und an den nachfolgenden Prozessschritt weitergegeben.

Die X4 BPMS verwendet Apache FOP als XSL-FO-Prozessor zum Verarbeiten der Reports.

Über das XSL-FO-Element `<fo:external-graphic src="url('Pfad/zur/Datei.png')">` können Grafikdateien (z. B. GIF, JPEG, PNG, SVG) ins Report-Dokument eingebunden werden. Um Grafiken aus dem Repository der X4 BPMS einzubinden, können Sie deren Pfad mit einer `xstore:/`-URL angeben, z. B.:

```
<fo:external-graphic src="url('xstore://Project/Folder/File.png')"
    content-width="10mm" content-height="10mm"/>
```

Mit den optionalen Attributen `content-width` und `content-height` können Sie Breite und Höhe des Bilds im Ausgabedokument bestimmen.

Alternativ können Sie binäre Grafikdateien auch als Base64-kodierte Zeichenkette in den Report nach folgendem Muster einbetten. Weitere Informationen hierzu finden Sie unter Base64 Converter:

```
<fo:external-graphic src="url(data:; base64,Base64-kodierte Bilddaten)"/>
```

Bei SVG-Bildern kann zudem deren Quellcode direkt in den Report eingebettet werden, z. B.:

```
<fo:instream-foreign-object>
   <svg:svg xmlns:svg="http://www.w3.org/2000/svg"
    height="1mm" width="40mm">
    <svg:line y1="0" y2="0" x1="0" x2="40mm"
     style="stroke:#0000FF;stroke-width:0.5pt" />
   </svg:svg>
</fo:instream-foreign-object>
```

Achten Sie hierbei darauf, dass alle SVG-Elemente den Namespace `svg:` besitzen.

Beispiel

Im Folgenden ein Beispiel für ein sehr einfaches (minimales) XSL-FO-Dokument:

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="simple" page-height="29.7cm"
    page-width="21cm" margin-left="2.5cm" margin-right="2.5cm">
      <fo:region-body margin-top="3cm"/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="simple">
    <fo:flow flow-name="xsl-region-body">
      <fo:block>Hello World</fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

Weitere Informationen zum Thema **Hyphenation** finden Sie unter https://xmlgraphics.apache.org/fop/2.5/hyphenation.html.

3.5 Elemente im Bereich „Drawing“

Folgende Zeichnungselemente können Sie in Technical Processes verwenden:

3.5.1 Connector

`Connector`: Ermöglicht die logische Verbindung von beliebigen Elementen.

Connector-Verbindungslinien ausschließlich zur Visualisierung von logischen Zusammenhängen!
Bei der Ausführung über den X4 Server werden ausschließlich Prozessbausteine in den Daten- und Kontrollfluss mit einbezogen, die in Business Processes über Sequenzfluss-Linien (siehe Sequence Flow) bzw. in Technical Processes über Transition-Übergangslinien verbunden sind (siehe Transition).

Eigenschaften

`Label`	Freitextfeld zur Beschriftung der Verbindungslinie (zur übersichtlicheren Gestaltung bzw. Dokumentation)
`Line Routing`	Linienführung **Mögliche Werte:** • *`Angled`*: Angewinkelte Linie • *`Straight`*: Gerade Linie (Standard)

`Line Type`	Linientyp **Mögliche Werte:** • *`Dash line`*: Gestrichelte Linie (Standard) • *`Solid line`*: Durchgezogene Linie • *`Dot line`*: Gepunktete Linie
`Line Color`	Farbe der Verbindungslinie. Um die Farbe zu ändern, die Eigenschaft markieren und auf klicken. **Mögliche Werte:** • *`RGB(RRR,GGG,BBB)`*: 24-bit-RGB-Farbcode (über den Farbauswahldialog wählbar) • *`RGB(0,0,0)`*: Schwarz (Standard)
`Line Source Decorator` und `Line Target Decorator`	Symbol für den Linienanfang **Mögliche Werte:** • *`Circle`*: Kreis (Standard) • *`Empty arrow`*: Ungefüllter Pfeil • *`Association arrow`*: Offener Pfeil • *`Arrow`*: Gefüllter Pfeil • *`Inverted arrow`*: Umgekehrter Pfeil • *`Diamond`*: Gefüllte Raute • *`Empty diamond`*: Ungefüllte Raute • *`Plain`*: Kein Symbol

3.5.2 Rectangle und Ellipse

Rectangle	Zeichnet ein beliebiges Rechteck
Ellipse	Zeichnet eine beliebige Ellipse

Eigenschaften

`Line Color`	Rahmenfarbe. Um die Farbe zu ändern, die Eigenschaft markieren und auf klicken. **Mögliche Werte:** • *`RGB(RRR,GGG,BBB)`*: 24-bit-RGB-Farbcode (über den Farbauswahldialog wählbar) • *`RGB(0,0,0)`*: Schwarz (Standard)
`Line Width`	Linienbreite des Zeichnungselements **Mögliche Werte:** • beliebige positive ganze Zahl • *`1`*: 1 Pixel breite Linie (Standard)
`Hide Line`	Rahmenlinie ausblenden
`Fill Color`	Füllfarbe des Zeichnungselements; Um die Farbe zu ändern, die Eigenschaft markieren und auf klicken. **Mögliche Werte:** • *`RGB(RRR,GGG,BBB)`*: 24-bit-RGB-Farbcode (über den Farbauswahldialog wählbar) • *`RGB(255, 255, 255)`*: Weiß (Standard)
`Transparent`	Zeichnungselement als transparent kennzeichnen. Durch Aktivieren der Eigenschaft wird die Eigenschaft `Fill Color` aufgehoben.

3.5.3 Textbox

`Textbox`: Fügt einen Textrahmen ein, dessen Inhalt Sie per Doppelklick in einem Dialogfenster bearbeiten und formatieren können.

Eigenschaften

`Hyperlink`	Hinterlegte Verlinkung für das Textfeld. Zum Verlinken eine gültige URL eintragen. Mit `Strg` + `Klick` lässt sich die eingetragene URL aufrufen.
`Text`	Inhalt des Textrahmens; um den Text zu ändern und zu formatieren, die Eigenschaft markieren und auf klicken.

`Rotation`	Drehung des Textrahmens **Mögliche Werte:** • *`0`*: keine Drehung (Standard) • *`90`*: um 90° gegen den Uhrzeigersinn drehen • *`180`*: um 180° drehen • *`270`*: um 270° gegen den Uhrzeigersinn drehen
`Fill Color`	Füllfarbe des Textfeldes. Um die Farbe zu ändern, die Eigenschaft markieren und auf klicken. **Mögliche Werte:** • *`RGB(RRR,GGG,BBB)`*: 24-Bit-RGB-Farbcode (über den Farbauswahldialog wählbar) • *`RGB(255, 255, 255)`*: Weiß (Standard)
`Transparent`	Textfeld als transparent setzen. Durch Aktivieren der Eigenschaft wird die Eigenschaft `Fill Color` aufgehoben.

3.5.4 Polygon

`Polygon`: Zeichnet ein Polygon mit beliebig vielen Eckpunkten.

Eigenschaften

`Line Color`	Rahmenfarbe. Um die Farbe zu ändern, die Eigenschaft markieren und auf klicken. **Mögliche Werte:** • *`RGB(RRR,GGG,BBB)`*: 24-Bit-RGB-Farbcode (über den Farbauswahldialog wählbar) • *`RGB(0,0,0)`*: Schwarz (Standard)
`Line Width`	Linienbreite des Zeichnungselements **Mögliche Werte:** • beliebige positive ganze Zahl • *`1`*: 1 Pixel breite Linie (Standard)

`Fill Color`	Füllfarbe des Zeichnungselements. Um die Farbe zu ändern, die Eigenschaft markieren und auf klicken. **Mögliche Werte:** • *`RGB(RRR,GGG,BBB)`*: 24-Bit-RGB-Farbcode (über den Farbauswahldialog wählbar) • *`RGB(255,255,255)`*: Weiß (Standard)
`Transparent`	Transparente Füllung (keine Füllfarbe verwenden). Durch Aktivieren der Eigenschaft wird die Eigenschaft `Fill Color` aufgehoben.

3.5.5 Polyline und Line

`Polyline`	Zeichnet eine Linie mit beliebig vielen Wegpunkten
`Line`	Zeichnet eine gerade Linie

Eigenschaften

`Line Color`	Linienfarbe. Um die Farbe zu ändern, die Eigenschaft markieren und auf klicken. **Mögliche Werte:** • *`RGB(RRR,GGG,BBB)`*: 24-Bit-RGB-Farbcode (über den Farbauswahldialog wählbar) • *`RGB(0,0,0)`*: Schwarz (Standard)
`Line Width`	Linienbreite des Zeichnungselements **Mögliche Werte:** • beliebige positive ganze Zahl • *`1`*: 1 Pixel breite Linie (Standard)

3.5.6 Image

`Image`: Fügt einen Bildrahmen ein, mit dem sich eine Grafikdatei vom Typ `.gif`, `.jpg`, `.png` oder `.bmp` einbinden lässt.

> Das eingefügte Bild wird im Prozess gespeichert.

Eigenschaften

`Line Color`	Linienfarbe des Bildrahmens. Um die Farbe zu ändern, die Eigenschaft markieren und auf klicken. **Mögliche Werte:** • *`RGB(RRR,GGG,BBB)`*: 24-Bit-RGB-Farbcode (über den Farbauswahldialog wählbar) • *`RGB(0,0,0)`*: Schwarz (Standard)
`Line Width`	Linienbreite des Zeichnungselements **Mögliche Werte:** • beliebige positive ganze Zahl • *`1`*: 1 Pixel breite Linie (Standard)
`Image File`	Pfad der verknüpften Grafikdatei

4 Mit ESB-Projekten arbeiten

Das X4 Repository ist in Projekten organisiert. Projekte besitzen eine Dateisystem-Struktur und können beliebige Ordner, Prozesse, Prozessbausteine und andere Dateien enthalten.

4.1 Templates

4.2 Die Struktur von ESB-Projekten

ESB-Projekte haben eine vordefinierte und nicht veränderbare Ordnerstruktur, die beim Anlegen eines neuen ESB-Projektes automatisch angelegt wird.

> ⓘ Die automatisch angelegten Ordner und Unterordner können nicht gelöscht, verschoben oder umbenannt werden.

`Adapters`	Hier können beliebig viele Adapter angelegt werden.
`Processes`	Hier können beliebig viele Technical Processes (`.wrf`) angelegt werden.
`Resources`	Hier können neben Ordnern auch Ressourcen wie Bilder, Text- und XML-Dokumente sowie weitere Dateien abgelegt werden.
`TemporaryFiles`	Hier werden temporär verwendete Dateien abgelegt.
`Transformations`	Hier können Transformationen (`.xsl`) sowie Reports (`.rep`) abgelegt werden
`<Projekt>.wsinc`	Webservice-Definition, die automatisch beim Anlegen des Projektes angelegt und nach dem Projekt benannt wird, z. B. *`ESBProject_Documentation.wsinc`*. ⓘ Ein Projekt kann lediglich eine Webservice-Definition enthalten.

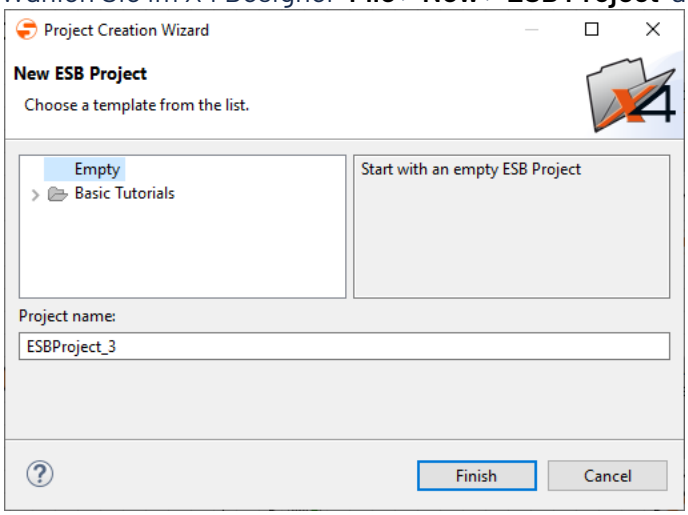
4.3 ESB-Projekt anlegen

Sie haben verschiedene Möglichkeiten, um ein BPM-Projekt im X4 Designer anzulegen:

- Über **New ESB Project** auf der Welcome-Seite des X4 Designers
- Über das Kontextmenü **New** > **ESB Project** in der Sicht **Projects**
- Über das Menü **File** > **New** > **ESB Project**

So legen Sie ein ESB-Projekt an

1. Wählen Sie im X4 Designer **File** > **New** > **ESB Project** aus.



2. Wählen Sie ein leeres Projekt oder eine vorhandene Projektvorlage aus.

   > Es können auch Projekte auf Basis eines Templates aus einer älteren Version der X4 Suite erstellt werden. Diese werden während des Erstellens automatisch migriert.

3. Geben Sie unter **Project name** den Namen des Projekts ein.
4. Klicken Sie auf **Finish**, um das Projekt anzulegen.
   In der Sicht **Projects** wird ein leeres ESB-Projekt mit der vordefinierten Struktur angelegt.
   Über das Kontextmenü können Sie nun Elemente für Ihr ESB-Projekt auswählen und im jeweiligen Ordner anlegen. Welche Optionen im Kontextmenü verfügbar sind, hängt davon ab, an welcher Stelle in der Baumstruktur das Kontextmenü aufgerufen wird.

> Wenn Sie beim Anlegen des Projekts den Projekttyp **Expert ESB Project** auswählen, können Sie ein leeres ESB-Projekt ohne vorgegebene Ordnerstruktur anlegen und im Expertenmodus arbeiten. Bei diesem Projekttyp können Sie die Ordnerstruktur und die darin enthaltenen Elemente selbst festlegen.
>
> **Hinweise zur Verwendung von ESB-Projekten aus älteren X4-Versionen:**
>
> - ESB-Projekte aus X4-Versionen vor 6.x werden automatisch in ein Expert ESB Project importiert.
> - *Classic ESB Projects* aus älteren X4-Versionen werden automatisch in ein Expert ESB Project umgewandelt.

4.4 Projekte verwalten

Erfahren Sie hier, wie Sie Projekte im Repository Navigator verwalten.

4.4.1 Repository-Elemente importieren

Hier erfahren Sie, wie Sie Projekte sowie verschiedene Ressourcen aus dem lokalen Dateisystem in die Sicht **Projects** importieren.

4.4.1.1 Importieren via Drag&Drop

Aus dem Dateisystem (z. B. vom Desktop oder dem *Windows Explorer*) können Sie eine oder mehrere markierte Dateien per Drag & Drop in die Sicht **Projects** und damit ins aktuell verbundene Repository importieren. Falls eine Datei gleichen Namens am Ziel bereits vorhanden ist, werden Sie aufgefordert, einen neuen Namen einzugeben.

Beachten Sie:

- Ordnerstrukturen werden beim Import nicht unterstützt, verwenden Sie hierzu den entsprechenden Wizard.
- Bei größeren Dateiübertragungen (insgesamt über 1 MB) wird ein Bestätigungsdialog eingeblendet, da möglicherweise Verzögerungen bei der Übertragung sowie Arbeitsspeicher-Probleme auftreten können.
- Importierte Dateien, die Sonder-/Leerzeichen oder Umlaute im Dateinamen enthalten, müssen zunächst umbenannt werden, bevor Sie diese im X4 Designer öffnen können.
- Projekte verfügen über eine vordefinierte Struktur, sodass die eingefügten Dateien nur im jeweils vordefinierten Ordner abgelegt werden können, siehe auch Die Struktur von ESB-Projekten

4.4.1.2 Import Wizard für Dateien und Projekte nutzen

Beachten Sie

- Projekte verfügen über eine vordefinierte Struktur, sodass die zu importierenden Dateien nur im jeweils vordefinierten Ordner abgelegt werden können, siehe auch Die Struktur von ESB-Projekten
- Projekte können lediglich auf oberster Ebene und in der vordefinierten Projektstruktur importiert werden, siehe auch Die Struktur von ESB-Projekten

1. Menü **Projekt > Import** aufrufen.
   Der **Resource Import Wizard** wird aufgerufen.
2. Ressource auswählen.
   - Über **Browse** die zu importierende Datei (`*.zip`) oder das zu importierende Projekt wählen und in **New project/file name** den entsprechenden Namen eingeben.
3. **Finish** klicken.

4.4.2 Repository-Elemente exportieren

Hier erfahren Sie, wie Sie Projekte sowie verschiedene Ressourcen aus der Sicht **Projects** ins Dateisystem exportieren.

4.4.2.1 Export Wizard für Dateien und Projekte nutzen

Beachten Sie

- Projekte können nur dann exportiert werden, wenn die oberste Projektebene ausgewählt ist.

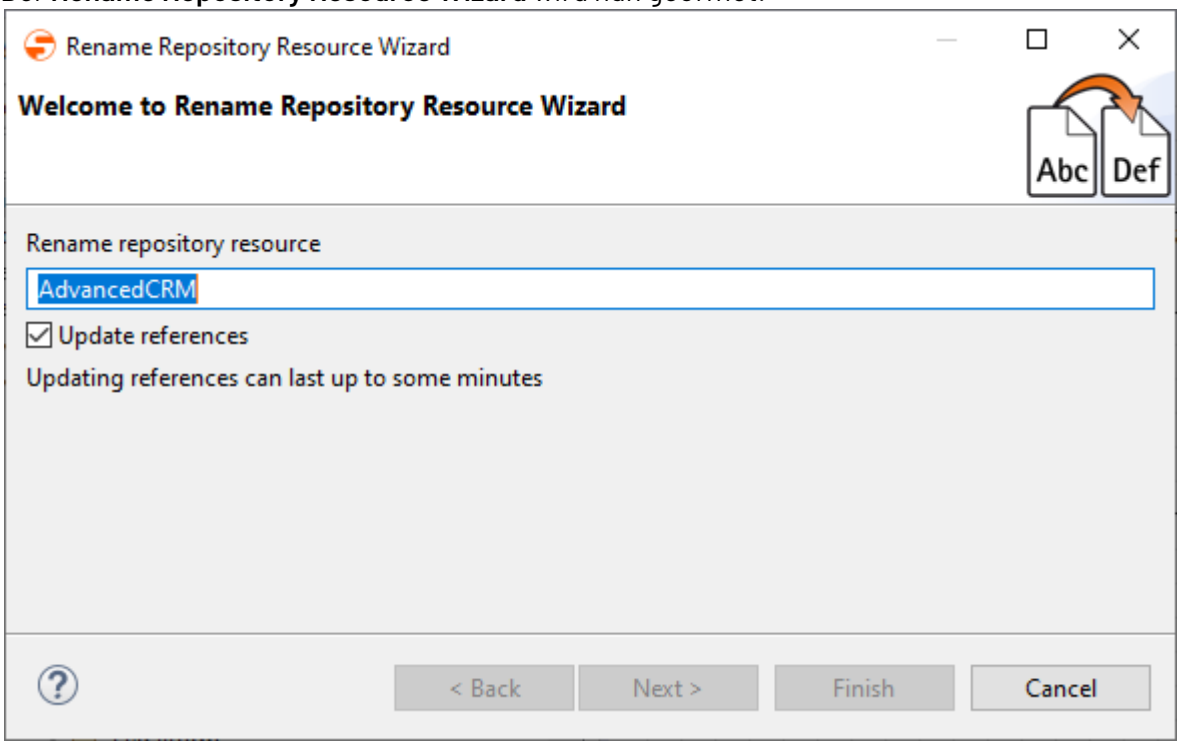
1. Menü **Projekt > Export** aufrufen.
   Der **Resource Export Wizard** wird aufgerufen.
2. Zielordner auswählen.
3. **Speichern** klicken.

4.4.3 Repository-Elemente intelligent umbenennen

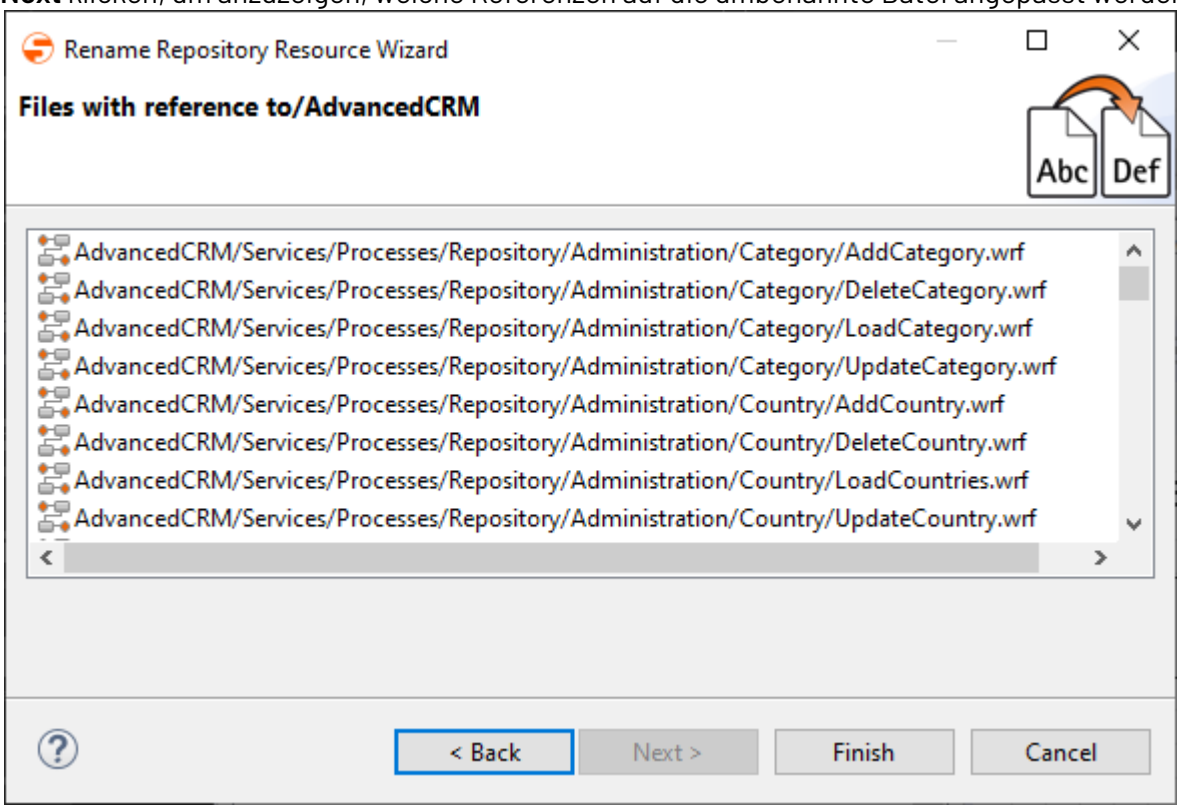
Wenn Sie Dateien im Repository umbenennen möchten, können Sie über den **Rename Repository Resource Wizard** die Referenzen in den Prozessen automatisch aktualisieren lassen.

1. Auf das gewünschte Element im Repository Navigator rechtsklicken, um das Kontextmenü zu öffnen.

2. **Refactor/Rename** wählen.
   Der **Rename Repository Resource Wizard** wird nun geöffnet.



3. In **Rename repository resource** einen neuen Dateinamen eingeben.
4. **Next** klicken, um anzuzeigen, welche Referenzen auf die umbenannte Datei angepasst werden.



5. **Finish** klicken, um die gewünschten Änderungen durchzuführen.
   Die betroffenen Dateien werden nun umbenannt und die entsprechenden Referenzen zu Prozessen, Ressourcen oder Komponenten angepasst.

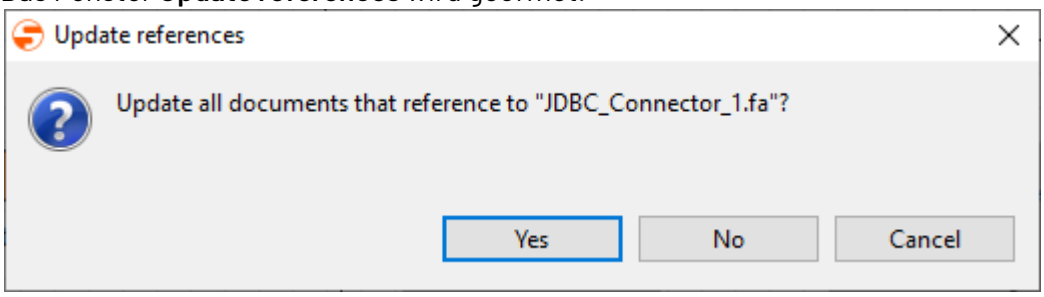
4.4.4 Repository-Elemente intelligent verschieben

Sie können Elemente innerhalb vom Repository verschieben. Dabei können die Referenzen auf die verschobene Datei in anderen Prozessen auf Wunsch automatisch angepasst werden.

1. Ressource an die gewünschte Stelle im Repository verschieben.

   > Projekte verfügen über eine vordefinierte Struktur, sodass die eingefügten Dateien nur im jeweils vordefinierten Ordner abgelegt werden können, siehe auch Die Struktur von ESB-Projekten

   Das Fenster **Update references** wird geöffnet.

2. Wenn sämtliche Referenzen auf die verschobene Datei bzw. den Ordner geändert werden sollen:
   - **Yes** klicken.
     Das Fenster **ConfirmMove** wird geöffnet. Hier wird angezeigt, von welchen Elementen im Repository die verschobene Datei verwendet wird.
   - **OK** klicken, um die gewünschten Änderungen durchzuführen.

     ? Unknown Attachment Beim Verschieben von XSL-Mappings mit referenzierten Input- und Output-XML-Dokumenten/Schemas müssen diese Dateireferenzen manuell angepasst werden, falls sich deren relative Pfade geändert haben.
3. Wenn die Referenzen nicht aktualisiert werden sollen: **No** klicken.

> **Tipp**
> Wenn Sie die **Strg**-/**Alt**-Tasten drücken, können Sie auch mehrere Dateien markieren und verschieben. Für jede Datei wird dann das Fenster **Update references** aufgerufen. Dort erhalten Sie zusätzlich folgende Möglichkeiten:
>
> - **Yes To All**: Referenzen für alle zu verschiebenden Dateien aktualisieren.
> - **No To All**: Alle Referenzen nicht aktualisieren.

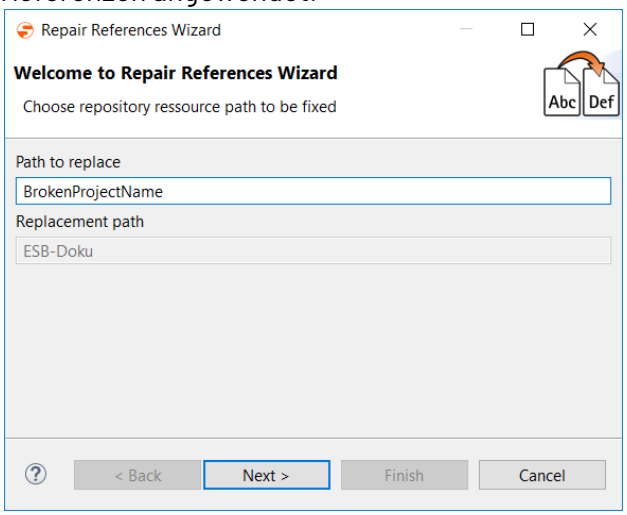
4.4.5 Repository-Referenzen reparieren

Wenn in einem Prozessdiagramm Repository-Elemente falsch referenziert sind – etwa weil ein Projekt oder Ordner umbenannt wurde oder von einer veralteten Vorlage erstellt wurde – so können Sie diese Referenzen über den **Repair References Wizard** reparieren lassen. Dabei werden die Pfadangaben für Prozessbausteine automatisch auf den tatsächlichen Pfad im Repository-Projekt angepasst.

1. Öffnen Sie einen Technical Process, Business Process oder Process Landscape in dem Referenzen auf Repository-Bausteine fehlerhaft (ausgegraut) sind und ermitteln Sie in der **Properties**-Sicht den inkorrekten Pfadbestandteil für einen markierten Prozessbaustein.

   > **Beispiel**
   > In einem Prozessdiagramm sind die Prozessbausteine inkorrekt referenziert: Deren Repository-Pfade verweisen auf das nicht vorhandene Projekt . Diese Pfadangaben sollen auf den tatsächlichen Projektnamen `ProjectName` korrigiert werden.

2. Auf das betroffene Projekt oder den betroffenen Ordner im Repository Navigator rechtsklicken, um das Kontextmenü zu öffnen.
3. **Repair References** wählen, um den **Repair References Wizard** aufzurufen.
   Im Feld **Replacement path** wird nun das markierte Projekt bzw. der gewünschte Ordnername angezeigt. Dieser Pfadbestandteil wird im nächsten Schritt automatisch auf die fehlerhafte Referenzen angewendet.



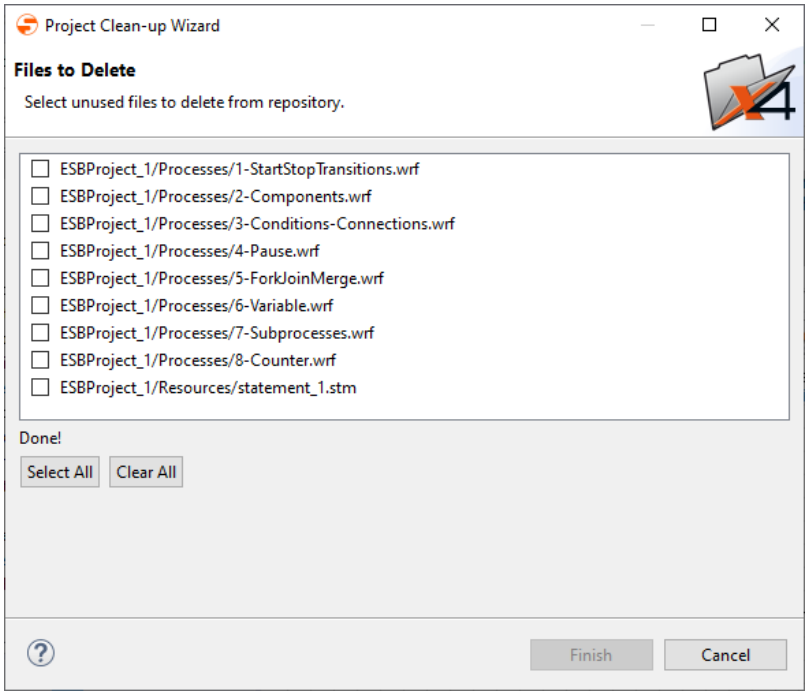
4. In **Path to replace** den fehlerhaften Projekt- oder Ordnernamen angeben, den Sie in Schritt 1 ermittelt haben.
5. **Next** klicken, um eine Liste von Prozessen anzuzeigen, welche die fehlerhaften Referenzen aufweisen und angepasst werden sollen.
6. **Finish** klicken, um die gewünschten Änderungen durchzuführen.

4.4.6 Nicht verwendete Ressourcen löschen

Wenn im Repository nicht verwendete Ressourcen vorhanden sind, so können Sie diese über den **Project Clean-Up Wizard** auflisten und auf Knopfdruck entfernen.

1. Auf das betroffene Projekt im Repository Navigator rechtsklicken, um das Kontextmenü zu öffnen.
2. **Clean-Up Project** wählen, um den **Project Clean-Up Wizard** aufzurufen.
   Alle nicht verwendeten Dateien werden nun aufgelistet.

   ⓘ Dieser Vorgang kann einige Zeit in Anspruch nehmen.



3. Gewünschte Ressourcen wählen.

   ⚠ **Achtung: Möglicher Datenverlust!**
   Beachten Sie, dass es möglich ist, dass Dateien angezeigt werden, die z. B. als dynamischer Parameter in einem XSL-Mapping referenziert werden.
   Löschen Sie nur Dateien, von denen Sie sicher sind, dass Sie diese nicht mehr benötigen!

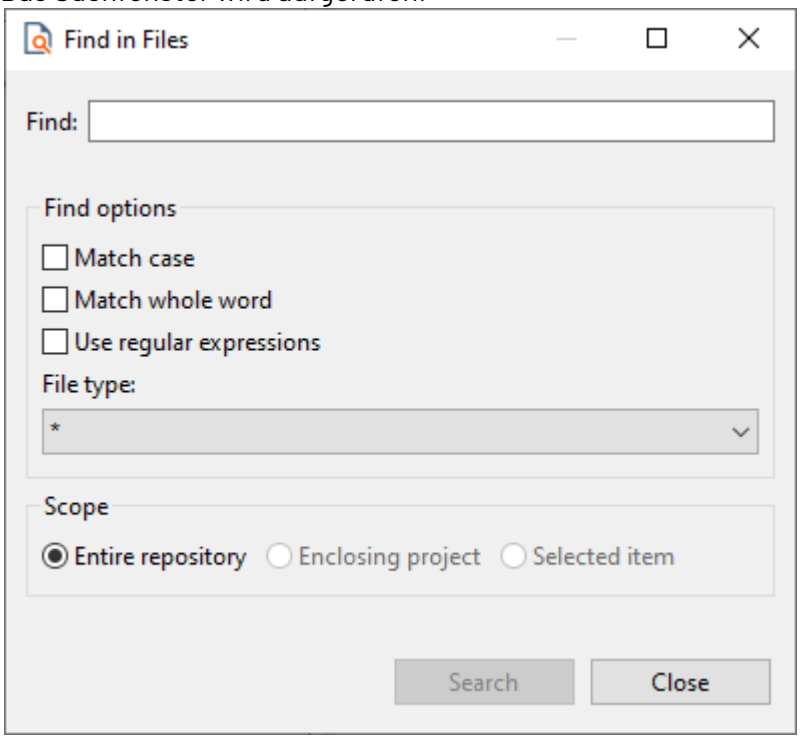
4. **Finish** klicken, und das endgültige Löschen anschließend mit **Yes** bestätigen.
   Die Ressourcen wurden nun gelöscht.

4.4.7 Repository-Elemente durchsuchen

Hier erfahren Sie, wie Sie innerhalb des Repository mit der Funktion **Find in Files** nach Texten, Regulären Ausdrücken und UTF-8-Symbolen suchen können.

Bei der Suche werden lediglich Dateiinhalte durchsucht. Dateinamen werden bei der Suche nicht berücksichtigt.

1. Menü **Edit** > **Find in Files** aufrufen.
   Das Suchfenster wird aufgerufen.



2. Bei **Find** Zeichen eingeben, nach dem gesucht werden soll.
3. Suchoptionen festlegen:
   - **Match Case:** Groß-/Kleinschreibung bei der Suche beachten.
   - **Match whole word:** Nach dem vollständigen Wort suchen.
   - **Use regular expression:** Nach regulären Ausdrücken suchen.
   - **File type:** Dateityp angeben, der durchsucht werden soll, z. B. `bpm` oder `fa`.

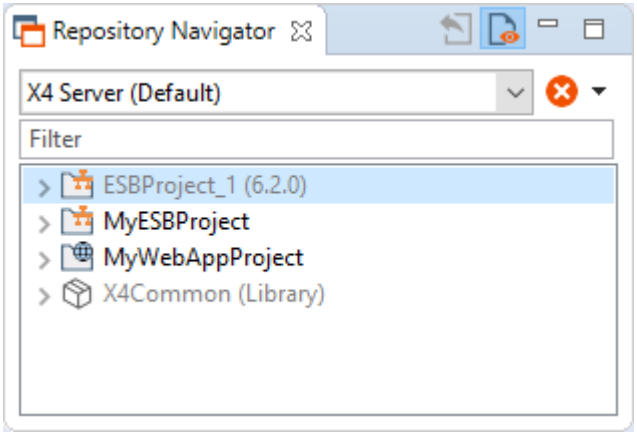
     Mit der Einstellung `*` werden alle Dateitypen durchsucht.
4. Umfang der Suche festlegen:
   - **Entire repository**: Im gesamten Repository suchen.
   - **Enclosing project:** Die Suche im ausgewählte Projekt ausführen.
   - **Selected item:** Suche auf ausgewählte Elemente beschränken.
5. **Search** klicken.

Sie können auch anhand von Adapter-Parametern suchen, sodass beispielsweise alle Adapter mit einem bestimmten Parameter angezeigt werden.

4.4.8 Veraltete Projekte migrieren

Veraltete Projekte werden im Repository Navigator besonders gekennzeichnet und werden lediglich in einer Read-only-Ansicht angezeigt. Um die Projekte weiter zu bearbeiten, bietet der X4 Designer eine Funktion zur Migration des Projektes in die aktuelle X4 BPMS-Version.

Dazu das gewünschte Projekt im Repository Navigator wählen und über das Kontextmenü **Migrate Project** oder das Menü **Project > Migrate Project** das Projekt migrieren.



⚠ Veraltete Projekte lassen sich ab Version 6.0.0 der X4 BPMS in die aktuelle X4 BPMS-Version migrieren.
Projekte mit einer Version älter als 6.0.0, müssen zunächst manuell auf Version 6.0.0 aktualisiert werden.

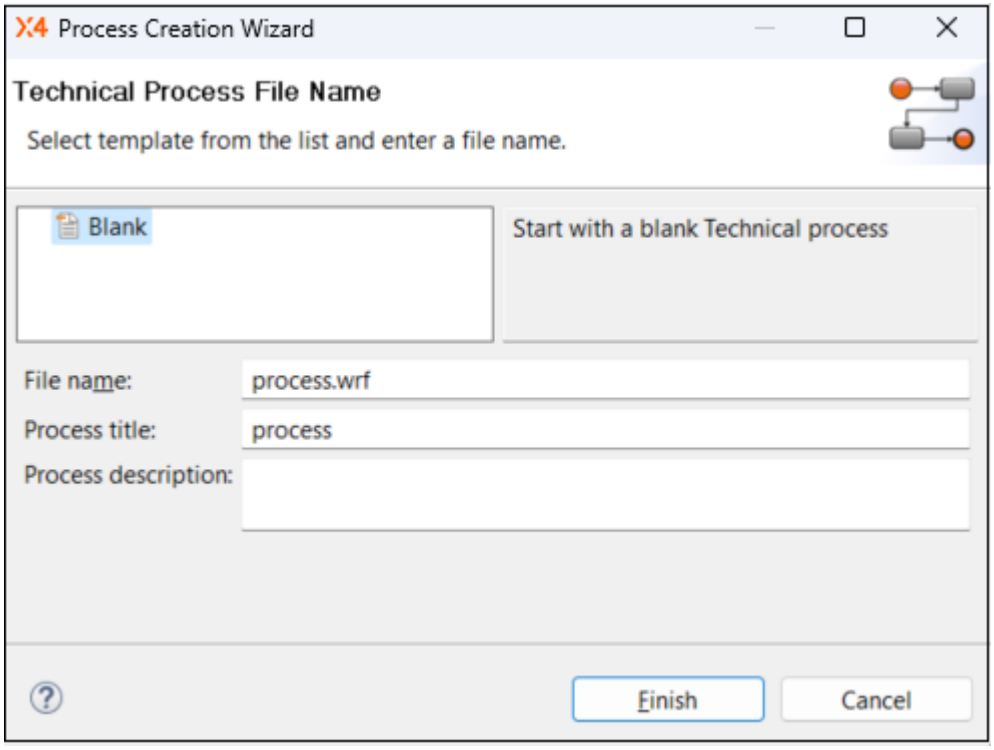
5 Prozesse anlegen und modellieren

5.1 Prozess anlegen

Mit dem Process Editor können Sie Technical Processes grafisch modellieren.
Sie können sowohl leere Bausteine aus der Palette als auch Dateien aus der Sicht **Projects** auf die Zeichnungsfläche ziehen und in der Sicht **Properties** die jeweiligen Eigenschaften konfigurieren, um einen ausführbaren Prozess zu erstellen.

Um einen neuen Prozess anzulegen, gehen Sie wie folgt vor:

1. Klicken Sie mit der rechten Maustaste auf den Ordner **Processes**, und wählen Sie im Kontextmenü **New > Technical Process...**.
   Das Dialogfenster **Process Creation Wizard** wird geöffnet.



2. Wählen Sie eine leere Prozessvorlage oder eine vorhandene Prozessvorlage.
3. Geben Sie im Feld **File name** den Prozessnamen ein.
   Leerzeichen, Umlaute und Sonderzeichen sind im Dateinamen nicht erlaubt!
4. Geben Sie im Feld **Process title** einen Prozesstitel ein.
   Der Prozesstitel wird im Prozessdiagramm angezeigt und kann in der Sicht **Properties** über die Eigenschaft `Title` verändert und bei Bedarf über die Eigenschaft `Show Title` ein- und ausgeblendet werden.
   Um die Eigenschaft `Show Title` anzuzeigen, klicken Sie in der Sicht **Properties** auf das Symbol **Show Advanced Properties**.
5. Hinterlegen Sie ggf. im Feld **Process description** eine Prozessbeschreibung. Die Angabe einer Prozessbeschreibung ist optional.

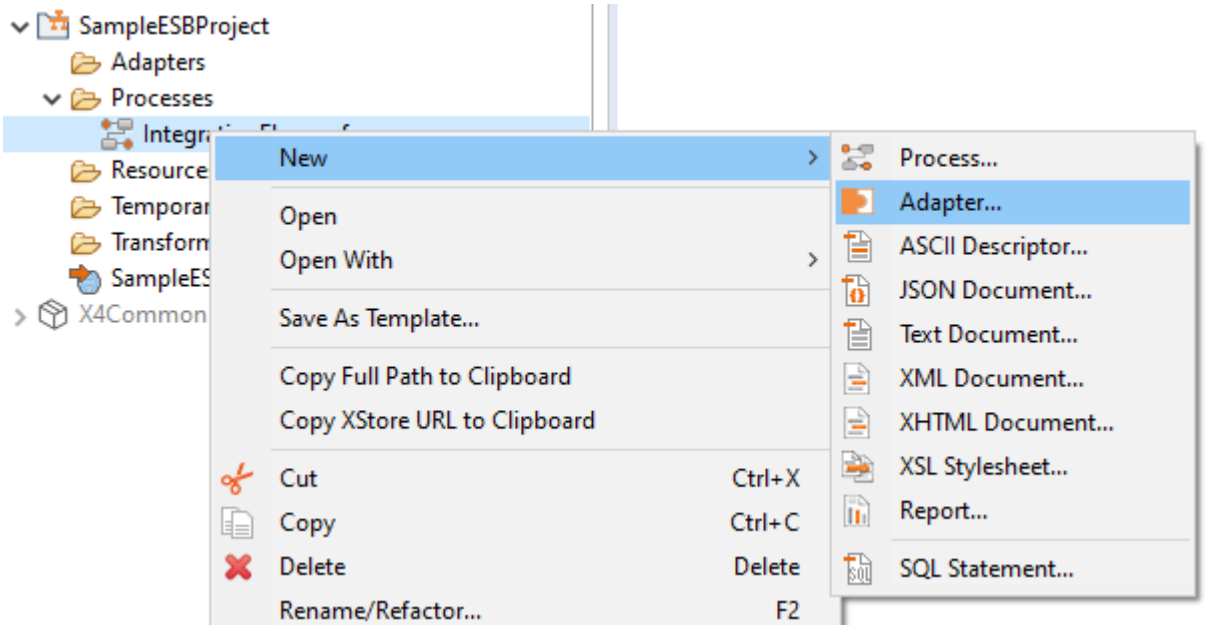
6. Wählen Sie **Finish**, um den Prozess anzulegen.
   Ein leeres Prozessdiagramm wird mit der Dateinamensendung `.wrf` in der Sicht **Projects** erstellt und im Process Editor geöffnet.
7. Modellieren Sie den Prozess wie gewünscht:
   - Ziehen Sie Bausteine aus der **Palette** oder Dateien aus der Sicht **Projects** per Drag&Drop auf eine bereits vorhandene Übergangslinie zwischen zwei Prozessbausteinen oder an eine freie Stelle auf der Zeichnungsfläche.
   - Zeichnen Sie ggf. Übergangslinien, um die Prozessbausteine zu verbinden. Gehen Sie dazu wie folgt vor:
     - Fahren Sie mit dem Mauszeiger über den Quellbaustein, um einen kleinen Pfeil anzuzeigen.
     - Klicken Sie auf den Pfeil, und ziehen Sie eine Linie zum Zielbaustein.

> **Tipp**
> Über das Menü **Project > Validate Project** lässt sich der Prozess über einen integrierten Prozess-Validator auf Gültigkeit prüfen.
> Wenn Sie Dateien aus der Sicht **Projects** in Ihren Prozess einbinden und diese in der Sicht **Properties** entsprechend konfigurieren, können Sie den modellierten Prozess über ▶ ausführen.

5.1.1 Nutzung von Private Files unterhalb von Prozessen

Wenn Prozessressourcen nur innerhalb eines Prozesses verwendet werden sollen, lassen sich diese direkt unterhalb eines Elternprozesses anlegen. Damit können sie ausschließlich im Prozesskontext verwendet werden und lassen sich aus anderen Prozessen auch nicht aufrufen.

Grundsätzlich lassen sich unterhalb eines einzelnen Prozesses die gleichen Ressourcen anlegen wie innerhalb eines Projektes – seien es Unterprozesse, Dokumente wie beispielsweise XML-Dateien, SQL-Statements oder auch Adapter.



Die zu verwendenden Ressourcen können sowohl über das Kontextmenü **New > <*Ressourcenname*>** als auch über die Symbolleiste **> Add <*Ressourcenname*>** angelegt werden.
Die angelegten Ressourcen lassen sich anschließend wie gewohnt parametrieren und per Drag & Drop aus dem Repository in das Prozessdiagramm einbauen.

5.2 Eigenschaften eines Technical Process (.wrf)

Technical Processes (`.wrf`) werden über den Process Editor grafisch modelliert und können auf dem X4 Server ausgeführt werden, wenn sämtliche Bausteine entsprechend parametriert und korrekt mit Repository-Dateien verknüpft wurden.

Die Eigenschaften jedes Technical Processes lassen sich in der **Properties**-Sicht des Process Editors bearbeiten, wenn im Process Editor ein freier Bereich der Zeichnungsfläche angeklickt wird.
Standardmäßig werden die gebräuchlichsten Eigenschaften angezeigt, über die Symbolleiste der Properties-Sicht lassen sich jedoch zusätzliche Einstellungen vornehmen:

- *Show Categories*: Eigenschaften nach Kategorien geordnet anzeigen
- *Show Advanced Properties*: Erweiterte Eigenschaften anzeigen

Eigenschaften

Documentation

`Title`	Freitextfeld für den Prozesstitel
`Show Title`	Prozesstitel durch Aktivieren bzw. Deaktivieren des Auswahlfeldes ein- und ausblenden
`Description`	Prozessbeschreibung; Kann bei Bedarf geändert werden
`Version`	Projektversion Wenn keine Version gesetzt ist, wird der Zeitstempel als Version genommen.
`Project Name`	Projektname; Kann bei Bedarf geändert werden
`Created`	Erstelldatum des Prozesses
`Created by`	Name des Prozess-Erstellers
`Last Update`	Datum der letzten Änderung
`Last Update by`	Name des letzten Bearbeiters
Metadata	
`Manage Metadata`	Metadaten verwalten, z. B. hinzufügen, löschen oder bearbeiten; Dazu die Eigenschaft markieren und auf klicken, siehe Prozess-Metadaten hinzufügen
Technical	
`Schema Declaration`	Schema für Input und Output des Prozesses
`Location`	Vollständiger relativer Pfad des Prozesses innerhalb des X4 Repositorys; Wird automatisch gesetzt.
`Can Stop`	Prozess darf während der Ausführung angehalten werden Diese Eigenschaft wird nicht an Subprozesse vererbt. **Mögliche Werte:** • *`True`*: für normale Prozesse (Standard) • *`False`*: für kritische Prozesse, die zu Ende laufen müssen
`Stop on error`	Im Fehlerfall die Prozess-Ausführung anhalten **Mögliche Werte:** • *`True`*: Ausführung abbrechen, wenn ein Fehler auftritt (Standard) • *`False`*: Fehlerhaften Prozess weiter ausführen (sofern möglich)

`In service`	Prozess aktiv oder inaktiv setzen. Werden inaktive Prozesse direkt ausgeführt, so erhalten sie die Prozess-ID *-1* . Subprozesse, also Prozesse, die in anderen Prozessen aufgerufen werden, können unabhängig von dieser Eigenschaft immer ausgeführt werden. **Mögliche Werte:** • *True*: Prozess kann ausgeführt werden (Standard) • *False*: Prozess kann nicht ausgeführt werden (nur als Subprozess)
`Schedule`	Schedule-Werkzeug, das zeitgesteuertes Ausführen von Prozessen auf dem X4 Server ohne X4 Designer ermöglicht; Um einen Zeitplan einzurichten, die Eigenschaft markieren und auf klicken. **Mögliche Werte:** • *Disabled*: Kein Zeitplan wurde festgelegt (Standard) • *Enabled*: Ein Zeitplan für die Ausführung des Prozesses wurde hinterlegt
`Instance limit`	Maximale Anzahl von Instanzen dieses Prozesses, die parallel ausgeführt werden dürfen **Beachten Sie!** `Instance limit` wird ignoriert, wenn der Prozesse als Subprozess aufgerufen wird. **Mögliche Werte:** • Beliebige positive ganze Zahl • *0*: Keine Begrenzung (Standard)
`Owner`	Freitextfeld zur Benennung eines Prozess-Verantwortlichen

5.3 Prozessbausteine ausrichten

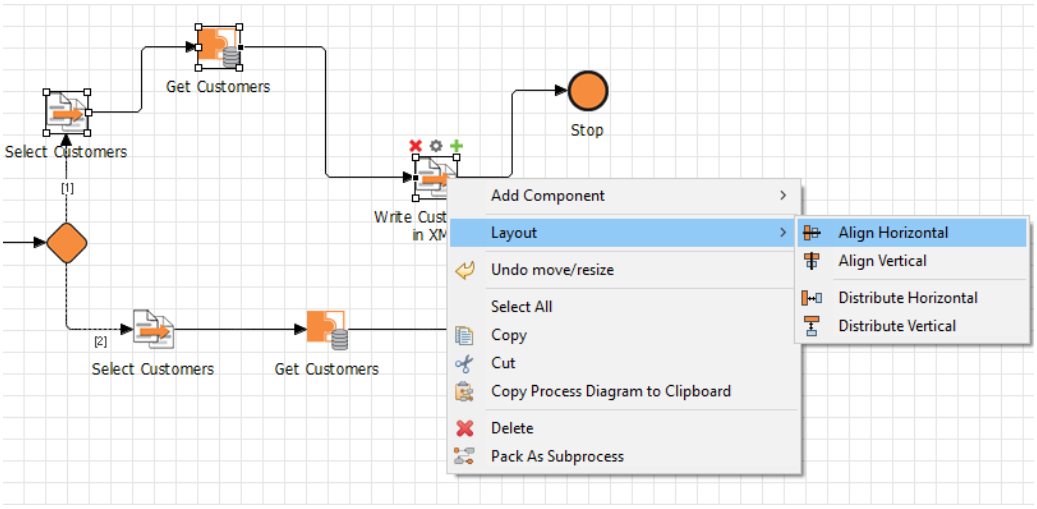
Prozessbausteine von Technical Processes und Business Processes können über den Kontextmenü-Eintrag **Layout** auf einfache Weise aneinander ausgerichtet werden. Damit schaffen Sie schnell und einfach eine saubere, strukturierte Prozessübersicht.

Dazu die gewünschten Prozesskomponenten auswählen und über Rechtsklick das Kontextmenü aufrufen. Anschließend **Layout** und die gewünschte Option wählen.

Folgende Optionen stehen für das Ausrichten der Prozesskomponenten zur Verfügung

- `Align Horizontal`: Komponenten anhand des Mittelwertes horizontal ausrichten
- `Align Vertical`: Komponenten anhand des Mittelwertes vertikal ausrichten
- `Distribute Horizontal`: Komponenten gleichmäßig horizontal verteilen

- `Distribute Vertical`: Komponenten gleichmäßig vertikal verteilen



> Für die Optionen `Align Horizontal` und `Align Vertical` müssen mindestens zwei Prozesskomponenten ausgewählt sein, für die Optionen`Distribute Horizontal` und `Distribute Vertical` mindestens drei Komponenten.

5.4 Prozesse validieren

Sowohl Business Processes (`.bpm`) als auch Technical Processes (`.wrf`) können über einen integrierten Prozess-Validator auf Gültigkeit geprüft werden. Dabei kann die Validierung auf Basis von standardmäßig hinterlegten oder benutzerdefinierten Validierungsregeln durchgeführt werden.

Die Prozess-Validierung kann über folgende Menüs geöffnet werden:

- Kontextmenü > **Validate** bei Rechtsklick auf Prozesse im Repository
- Menü **Project** > **Validate Project**

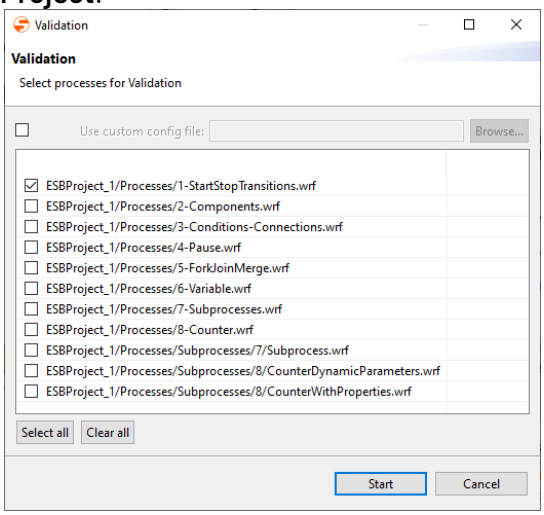
Bei der Prozess-Validierung werden drei Schweregrade unterschieden:

- `Error`: Fehler, welche die Ausführung des Prozesses verhindern
- `Warning`: Fehler, die eine fehlerhafte Ausführung des Prozesses verursachen können
- `Information`: Informationen zum Zustand eines Prozesses

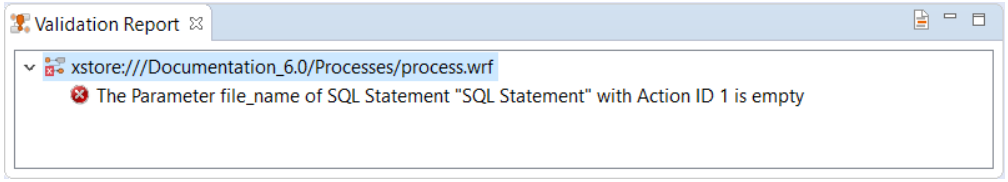
5.4.1 Prozess-Validierung durchführen

Um einen Prozess auf Gültigkeit zu prüfen, gehen Sie wie folgt vor:

1. Öffnen Sie den Dialog für die Prozess-Validierung, z.B. über das Menü **Project** > **Validate Project**.



2. Die zu validierenden Prozesse auswählen.
3. Über **Use custom config file** und **Browse**, ggf. eine Konfiguration für eigene Validierungsregeln hochladen, siehe Abschnitt *Benutzerdefinierte Validierungsregeln*.
4. Mit **Start** die Validierung beginnen.
   In der **Validation Report** Sicht wird nun das Validerungsergebnis geöffnet.



- Mit Doppelklick auf den Prozesspfad wird der entsprechende Prozess im X4 Designer geöffnet.
- Mit Doppelklick auf ein Validierungsergebnis werden die Eigenschaften der entsprechende Prozesskomponente in der *Properties-Sicht* geöffnet.

5.4.2 Standardmäßig hinterlegte Validierungsregeln

Folgende Validierungsregeln sich bereits standardmäßig hinterlegt:

Validierungsregel	Beschreibung	Schwere
`Action Parameter Validation`	Sind alle Aktionen korrekt parametriert?	Error
`Condition Validation`	Sind alle Zweige eines `Condition`-Baust eine gefüllt?	Warning
`Connection Validation`	Sind alle Pfade geschlossen?	Error

5.4.3 Benutzerdefinierte Validierungsregeln

Es können auch eigene Validierungsregeln definiert werden. Diese müssen als XSLT-Mapping mit einer vorgegebenen Struktur vorliegen.

Beispiel: Mapping-Beispiel für benutzerdefinierte Validierungsregeln

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
  <xsl:output media-type="text/xml" method="xml"></xsl:output>
  <xsl:template match="/">
    <xsl:param name="subprocessCount" select="count(Workflow/Actions/Action[Module/
text() = 'iXworkflow'])"></xsl:param>
    <xsl:param name="componentCount" select="count(Workflow/Actions/Action)"></xsl:pa
ram>
    <xsl:param name="textCount" select="count(Workflow/Graphics/svg/text)"></xsl:para
m>
    <step>
      <xsl:choose>
        <xsl:when test="$componentCount &gt; 100">
          <warn>The number of process components is <xsl:value-of select="$componentC
ount"></xsl:value-of>. Processes with more than 100 components may probably be
unclear and error-prone. Please check, if you can outsource parts of the process as
subprocesses.</warn>
        </xsl:when>
        <xsl:otherwise>
          <info>The number of process components is <xsl:value-of select="$componentC
ount"></xsl:value-of>
          </info>
        </xsl:otherwise>
      </xsl:choose>
      <xsl:choose>
        <xsl:when test="$textCount &gt; 0">
          <xsl:if test="$componentCount div $textCount &gt; 5">
            <warn>The ratio of process components to text components is greater than
5:1. It is <xsl:value-of select="$componentCount div $textCount"></xsl:value-of>:1.</
warn>
          </xsl:if>
        </xsl:when>
        <xsl:otherwise>
          <info>The process doesn't contain any text components.</info>
        </xsl:otherwise>
      </xsl:choose>
      <xsl:choose>
        <xsl:when test="$subprocessCount &gt; 0">
          <xsl:if test="( $componentCount - $subprocessCount ) div $subprocessCount
&gt; 10">
            <warn>The ratio of process components to subprocesses is greater than
10:1. It is <xsl:value-of select="( $componentCount - $subprocessCount ) div
$subprocessCount"></xsl:value-of>:1</warn>
          </xsl:if>
        </xsl:when>
        <xsl:otherwise>
          <info>The process doesn't contain any subprocesses.</info>
        </xsl:otherwise>
      </xsl:choose>
    </step>
  </xsl:template>
</xsl:stylesheet>
```

Für die Prozess-Validierung muss eine XML-Datei mit folgendem Aufbau übergeben werden. Das zu verwendende Mapping wird hier anhand der `xstore`-URL referenziert:

```
<validationConfig>
    <default active="true"/>
    <custom_xslt active="true" name="processMetrics">
        xstore://Projekt/Unterverzeichnis/customValidationMetrics.xsl
    </custom_xslt>
</validationConfig>
```

5.5 Prozessbausteine parametrieren

Auf den nachfolgenden Seiten erfahren Sie Folgendes:

- welche Arten von Parametern genutzt werden können
- wie Bausteine von Technical Processes parametriert bzw. konfiguriert werden
- wie Parameter innerhalb eines Technical Processes verwendet werden

Lesen Sie mehr dazu auf den folgenden Unterseiten:

- Möglichkeiten der Parametrierung
- Anwendungsbeispiele für Parameter
- Dynamische Parameter verwenden
- Component-Parameter verwenden
- Status mit ConditionState dynamisch setzen
- Parameter in XSL-Mappings verwenden

5.5.1 Möglichkeiten der Parametrierung

Hier erhalten Sie einen Überblick über die Erstellung und Verwendung von Parametern in der X4 BPMS.

Gültigkeitsbereich (Scope) für Parameter:

- **System-Parameter**: vom System vorgegebene Parameter (siehe System-Parameter)
- **Globale Parameter**: Parameter, die Sie als Nutzer anlegen können und die in allen Projekten der X4 BPMS verwendet werden können (siehe Globale Parameter)
- **Projekt-Parameter**: Parameter, die in einer Datei in einem Projekt erstellt, innerhalb dieses Projekts genutzt und mit dem Projekt importiert bzw. exportiert werden können (siehe Projekt-Parameter erstellen)

Verwendung von Parametern:

Parameter können in Technical Processes in Prozessbausteinen, Mappings, Adaptern und Conditions genutzt werden. In Geschäftsprozessen (BPM-Prozessen) können Sie zudem in Rules und Human Task verwendet werden. Es können globale, System- oder Projekt-Parameter verwendet werden.

Mit der Tastenkombination `Strg+Leertaste` können in Projektdateien (Prozessbausteinen, Mappings, Adaptern und Conditions) alle verfügbaren System-, Projekt- und globalen Parameter angezeigt und ausgewählt werden.

Beachten Sie folgende Priorisierung bei der Verwendung von Parametern:
System-Parameter → Projekt-Parameter → Globale Parameter
Bei Namenskonflikten wird eine Fehlermeldung bzw. Warnung in der Sicht **Problems** angezeigt:

- Fehlermeldung bei Namenskonflikten von System- und Projekt-Parametern
- Warnung bei Namenskonflikten von Projekt-Parametern und globalen Parametern

Wo werden Parameter gespeichert?

System-Parameter	Die X4 BPMS enthält eine Reihe von vordefinierten System-Parametern, z. B. `%PID%` (aktuelle Prozessinstanz-ID) oder `%NOW%` (aktuelles Datum und Uhrzeit). System-Parameter können nicht während der Laufzeit geändert werden. **Typischer Anwendungsfall:** Technische Parameter, z.B. eine Mailserver-Adresse, setzen für diese Laufzeitumgebung/Stufe. Dateinamen oder Prozess-ID eines Technical Processes ausgeben
Für zusätzliche Anwendungsfälle, die nicht durch System-Parameter abgedeckt werden, können Sie globale oder Projekt-Parameter erstellen. Globale oder Projekt-Parameter können, im Gegensatz zu System-Parametern, zur Prozesslaufzeit gesetzt werden.	
Globale Parameter	Globale Parameter können Sie in der **Global Parameter Configuration** definieren. Diese Konfiguration wird im jeweiligen Prozess gespeichert. Beispiele für Anwendungsfälle von globalen Parametern finden Sie in Globale Parameter verwenden. **Typischer Anwendungsfall:** Parameter setzen, die in verschiedenen Projekten genutzt und die während der Laufzeit geändert werden können, z.B. die Empfänger von systemgenerierten Mails oder die Fußzeile in systemgenerierten Mails.

Projekt-Parameter	Projekt-Parameter können in einem Projekt in mehreren Technical Processes verwendet werden. Diese Parameter werden in der **Project Parameter Configuration** (`*.param`) im Projekt gespeichert und können mit dem Projekt exportiert bzw. importiert werden. **Typischer Anwendungsfall:** • Technical Process: Ein Projekt-Parameter kann als Grenzwert für Messwerte (Messwert muss zwischen zwei Grenzwerten liegen) definiert werden. • Geschäftsprozess mit Business Rule: Ein Projekt-Parameter kann als Limitierung zur Überprüfung eines Werts (Abnahme/Ablehnung je nach gesetztem Wert) festgelegt werden. • Geschäftsprozess mit Human Task: Ein Projekt-Parameter kann in einem Mapping in einem Technical Process gesetzt werden, der vom Geschäftsprozess aufgerufen wird. Hier kann z.B. die Duration im Human Task unterschiedlich sein je nach Umgebung.

Wo werden Parameter verwendet?

In Technical Processes können Parameter für folgende Elemente gesetzt werden:

- Adapter
- Conditions
- Mappings
- weitere Prozessbausteine

In Geschäftsprozessen (BPM-Processes) können Parameter für folgende Elemente gesetzt werden

- Rules
- Human Task (über Mapping in einem Technical Process)

Beachten Sie folgende Priorisierung bei der Nutzung von **Component Parameters** und dynamischen Parametern:

- **Component Parameters** überschreiben die Konfiguration von Prozessbausteinen im Repository (beispielsweise Funktions-Adapter) während der Prozesslaufzeit.
- Dynamische Parameter überschreiben gesetzte globale, Projekt-Parameter System-Parameter sowie die Konfiguration von Prozessbausteinen im Repository (beispielsweise Funktions-Adapter) zur Prozesslaufzeit.

Aktion	Beschreibung
Adapter-Parameter	Parameter lassen sich in Adapter-Konfigurationen einsetzen. **Typischer Anwendungsfall:** In einem **XML Encryption (Password)** Adapter kann im Parameter xpath ein Pfad zu einem Element, das verschlüsselt/entschlüsselt werden soll, gesetzt werden. Dieser Pfad kann fest definiert (`//Data`) werden oder über einen Parameter gesetzt werden.
Conditions	Fallunterscheidungen in Verzweigungen von **Condition**-Bausteinen können Sie mit Parametern definieren (siehe Bedingungen erstellen und verwenden). **Typischer Anwendungsfall:** Es soll geprüft werden, ob der Output eines Prozessbausteins einen bestimmten Parameterwert enthält.
Mappings	Parameter (globale, Projekt- und System-Parameter) können in XSL-Mappings verwendet werden. Hierzu müssen Sie im Mapping eingefügt werden (siehe Parameter in XSL-Mappings verwenden).
Component-Parameter	Markierte Prozessbausteine eines geöffneten Prozesses können Sie in der Sicht **Properties** anpassen. Mit **Component-Parameter** können Sie einen markierten Prozessbaustein für den einzelnen Prozess parametrieren (siehe Component-Parameter). Diese Parameter werden im jeweiligen Prozess gespeichert. **Typischer Anwendungsfall:** Mit **Component-Parameter** können die Parameter eines Adapters in einem Prozess überschrieben werden. Innerhalb von **Component-Parameter** können Sie feste Werte (fester Pfad z.B. c:\temp) oder Parameter verwenden. Hier können System-Parameter oder zuvor definierte globale bzw. Projekt-Parametern genutzt werden.

Aktion	Beschreibung
Dynamische Parameter	Mit dynamischen Parametern können Sie globale, System- und Projekt-Parameter in XSL-Mappings überschreiben. Innerhalb eines Prozesses können Sie in Abhängigkeit von verarbeiteten Daten mit dynamischen Parametern verschiedene Prozessbausteine ansteuern. Dynamische Parameter werden in XSL-Mappings in Form von Processing Instructions abgelegt und vom darauf folgenden Prozessbaustein (z. B. einem Adapter) interpretiert (siehe Dynamische Parameter verwenden). Die Konfiguration wird nur während der Prozesslaufzeit dynamisch verändert. **Typischer Anwendungsfall:** Mit dynamischen Parametern können Sie Prozessbausteine ansteuern und bereits gesetzte Parameter zur Prozesslaufzeit überschreiben.

5.5.2 Anwendungsbeispiele für Parameter

Hier finden Sie mögliche Anwendungsbeispiele für die Verwendung von globalen, System- und Projekt-Parametern.

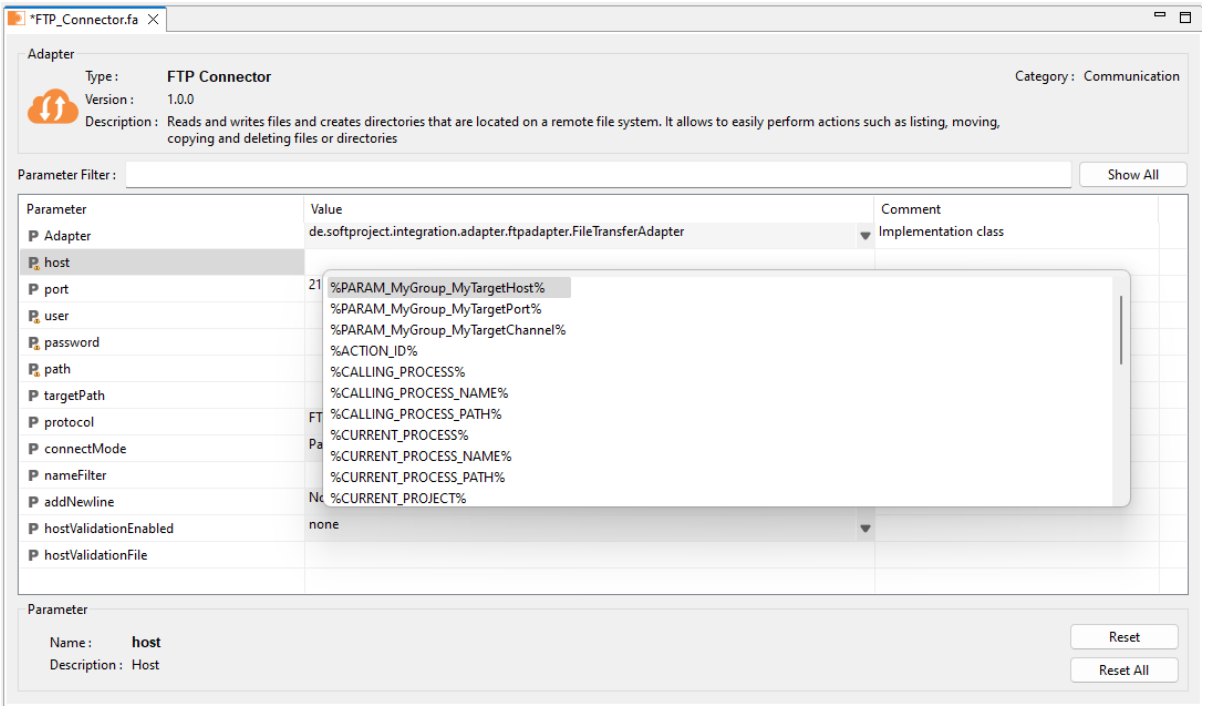
Parameter in Adaptern verwenden

Parameter können in Adaptern auf zwei Arten gesetzt werden:

- in der **Properties**-Sicht über **Component Parameters** (siehe Component-Parameter)
- in allen Textfeldern im Adapter-Editor

> **Hinweis**
> Bei Adaptern, werden in der Spalte `Value` mit der Tastenkombination `Strg` + `Leertaste` werd en alle verfügbaren System-Parameter sowie alle globalen und Projekt-Parameter vom Typ **String** angezeigt.

Der gewünschte Parameter kann aus einer Drop-down-Liste ausgewählt werden:



5.5.3 Dynamische Parameter verwenden

Wie kann ich dynamische Parameter verwenden?

Innerhalb eines Technical Processes können Sie in Abhängigkeit von verarbeiteten Daten mit sogenannten dynamischen Parametern Prozessbausteine ansteuern und ggf. bereits gesetzte Parameter zur Prozesslaufzeit überschreiben.

Dynamische Parameter werden in XSL-Mappings als Processing Instruction erzeugt und vom darauf folgenden Prozessbaustein (beispielsweise einem Adapter) verarbeitet.

Hierbei haben Sie mehrere Möglichkeiten:

- Parameter via XSL-Mapping setzen: Hier können Sie ein einzelnes XSL-Daten-Mapping einsetzen, das XML-Daten transformiert und gleichzeitig dynamisch Processing Instruction-Definitionen enthält.
- Dynamisch parametrieren ohne Datentransformation: Sie können auch ein separates XSL-Mapping erstellen, das nur die Processing Instructions generiert.
- Dynamisch parametrieren mit Processing Instructions und Dynamic Parameter: Sie können auch ein separates XSL-Mapping erstellen, das die Processing Instruction-Definitionen mit dynamischen Parametern enthält.

Hinweis
Beachten Sie folgende Priorisierung bei der Nutzung von **Component-Parameter** und dynamischen Parametern:

- **Component-Parameter** überschreiben die Konfiguration von Prozessbausteinen im Repository (beispielsweise Funktions-Adapter) während der Prozesslaufzeit.
- Dynamische Parameter überschreiben gesetzte globale, Projekt- und System-Parameter sowie die Konfiguration von Prozessbausteinen im Repository (beispielsweise Funktions-Adapter) zur Prozesslaufzeit.

5.5.3.1 Parameter dynamisch via XSL-Mapping setzen

1. Positionieren Sie ein XSL-Mapping im Technical Process vor dem Adapter oder der Datenbank-Verbindung.
2. Definieren Sie die Processing Instruction im XSL-Mapping nach folgendem Muster: `<xsl:processing-instruction name="DynamicParameter">`*`ParameterName`*`="`*`Parameter Value`*`"</xsl:processing-instruction>`
3. Ersetzen Sie `ParameterName` durch den gewünschten Parameternamen.
4. Ersetzen Sie `ParameterValue` durch den gewünschten Wert, meist in Form eines XSL-Befehls.
5. Wenn das XSL-Mapping ausschließlich dynamische Parameter erzeugen soll: Lesen Sie zusätzlich mit `<xsl:processing-instruction name="NoResult"/>` fest, dass das XSL-Mapping kein Ergebnis-Dokument erzeugt.

Hinweis
Das XSL-Mapping muss in jedem Fall ein Wurzel-Element generieren, damit das XSL-Mapping ausgeführt werden kann.

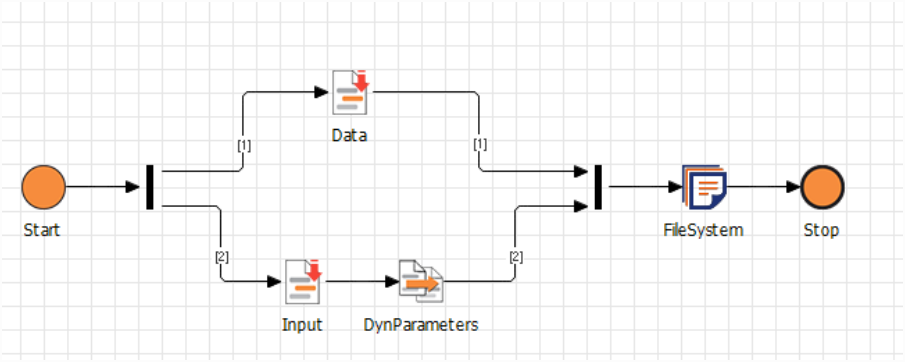
5.5.3.2 Dynamisch parametrieren ohne Datentransformation

Wenn ein Input-Dokument direkt an einen dynamisch parametrierten Adapter übergeben werden soll, ist es erforderlich, eine Gabelung (mithilfe des `Fork`-Bausteins) in das Diagramm des Technical Processes einzufügen:

1. Setzen Sie ein XSL-Mapping in einen Ausführzweig.
2. Setzen Sie den zu übergebenden Prozessbaustein in einen zweiten Ausführzweig.
3. Setzen Sie den Adapter direkt hinter den `Join`-Baustein.

Beispiel

Ein aus einem XML-Dokument ausgelesener Dateiname soll als dynamischer Parameter an einen Filesystem Transfer Adapter übergeben werden:



Im Beispiel besteht das XSL-Mapping aus folgendem Quelltext:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/" version="1.1">
  <xsl:output method="xml" media-type="text/xml"/>
  <xsl:template match="/">
    <xsl:processing-instruction name="DynamicParameter">documentname="
      <xsl:value-of select="data"/>"
    </xsl:processing-instruction>
    <xsl:processing-instruction name="NoResult"/>
    <root/>
 </xsl:template>
</xsl:stylesheet>
```

5.5.3.3 Dynamische Parameter mit DynamicParameter

Mit XSL-Mappings können Sie die Konfiguration von Adaptern und Datenbank-Verbindungen zur Laufzeit beeinflussen. Innerhalb des XSL-Mappings wird eine Logik implementiert, die in Abhängigkeit von den verarbeiteten Daten benutzerdefinierte Parameterwerte ausgibt. Diese benutzerdefinierten Parameterwerte können von einem nachfolgenden Adapter-, Transfer-Adapter- oder Datenbank-Baustein verarbeitet werden und überschreiben bereits gesetzte Werte.

Parameterwerte erzeugen

Um dynamische Werte innerhalb von Technical Processes zu setzen, können Sie mithilfe von XSL-Mappings Processing Instructions erzeugen, die an den nachfolgenden Prozess-Schritt übergeben werden und entsprechend interpretiert werden können.

```
<xsl:processing-instruction name="DynamicParameter">
parameterName="parameterValue"</xsl:processing-instruction>
```

Die Ausgabe des XSL-Mappings:

```
<?DynamicParameter parameterName="parameterValue" ?>
```

Parameter und Parameterwerte

Im XSL-Mapping können Sie beliebig viele unterschiedliche dynamische Parameter mit jeweils einem zugehörigen Wert definieren:

- `parameterName`: Zu überschreibender Parametername
- `parameterValue`: Zu überschreibender Parameterwert; immer innerhalb von Anführungszeichen

> **Hinweis**
> Beachten Sie, dass das XSL-Mapping ein Wurzel-Element generieren muss, damit das XSL-Mapping ausgeführt und ein gültiges XML-Dokument erzeugt werden kann.

Beispiel

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
  <xsl:output encoding="UTF-8" method="xml" indent="yes" version="1.0"/>
  <xsl:template match="/">
    <xsl:processing-instruction
    name="DynamicParameter">key="i"</xsl:processing-instruction>
    <root/>
  </xsl:template>
</xsl:stylesheet>
```

5.5.4 Component-Parameter verwenden

Wie kann ich Component-Parameter verwenden?

Für bestimmte Prozessbausteine (XSL-Mappings, PDF-Creator-Bausteine und Adapter) lassen sich **Component-Parameter** definieren.

Als **Component-Parameter** können Sie Parameter verwenden, die zur Laufzeit des Prozesses gesetzt werden.

Mögliche Parameter sind:

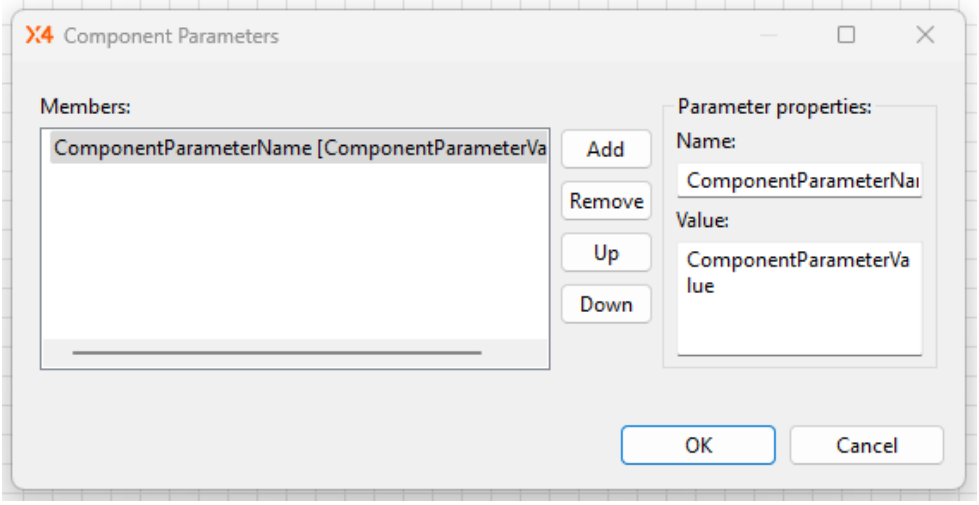
- System-Parameter, z. B. `%PID%` (aktuelle Prozessinstanz-ID) oder `%NOW%` (aktuelles Datum und Uhrzeit) (siehe Übersicht der System-Parameter)
- globale Parameter (siehe Globale Parameter und -Gruppen erstellen)
- Projekt-Parameter (siehe Projekt-Parameter erstellen)

> **Hinweis**
> Beachten Sie folgende Priorisierung bei der Nutzung von **Component-Parameter** und dynamischen Parametern:
>
> - **Component-Parameter** überschreiben die Konfiguration von Prozessbausteinen im Repository (beispielsweise Funktions-Adapter) während der Prozesslaufzeit.
> - Dynamische Parameter überschreiben gesetzte globale, Projekt- und System-Parameter sowie die Konfiguration von Prozessbausteinen im Repository (beispielsweise Funktions-Adapter) zur Prozesslaufzeit.

5.5.4.1 Component Parameter definieren

Um **Component-Parameter** zu definieren, gehen Sie wie folgt vor:

1. Rufen Sie das Fenster **Component-Parameters** auf:
   - Markieren Sie den Prozessbaustein im Prozess.
   - Klicken Sie in der **Properties**-Sicht auf `Component Parameters`, um anzuzeigen.
   - Klicken Sie auf , um das Fenster **Component-Parameters** aufzurufen.



2. Geben Sie in **Name** den Namen des **Component-Parameter** ein.
   - In XSL-Mappings und Reports können Sie den hier gesetzten Namen als Parameter ansprechen und den zugehörigen Wert übergeben, z. B. einen globalen Parameter `PDFPath` mit `<xsl:param name="PDFPath"/>`.
   - Verwenden Sie bei Adaptern und Datenbankverbindungen die exakte (technische) Bezeichnung des zu überschreibenden Parameters.
3. Geben Sie in **Value** den zu überschreibenden Parameterwert ein.

   > **Hinweis**
   > Mit `Strg` + `Leertaste` werden alle verfügbaren globalen, System- und Projekt-Parameter angezeigt.

4. Klicken Sie **OK**, um den **Component-Parameter** im Prozess zu speichern.

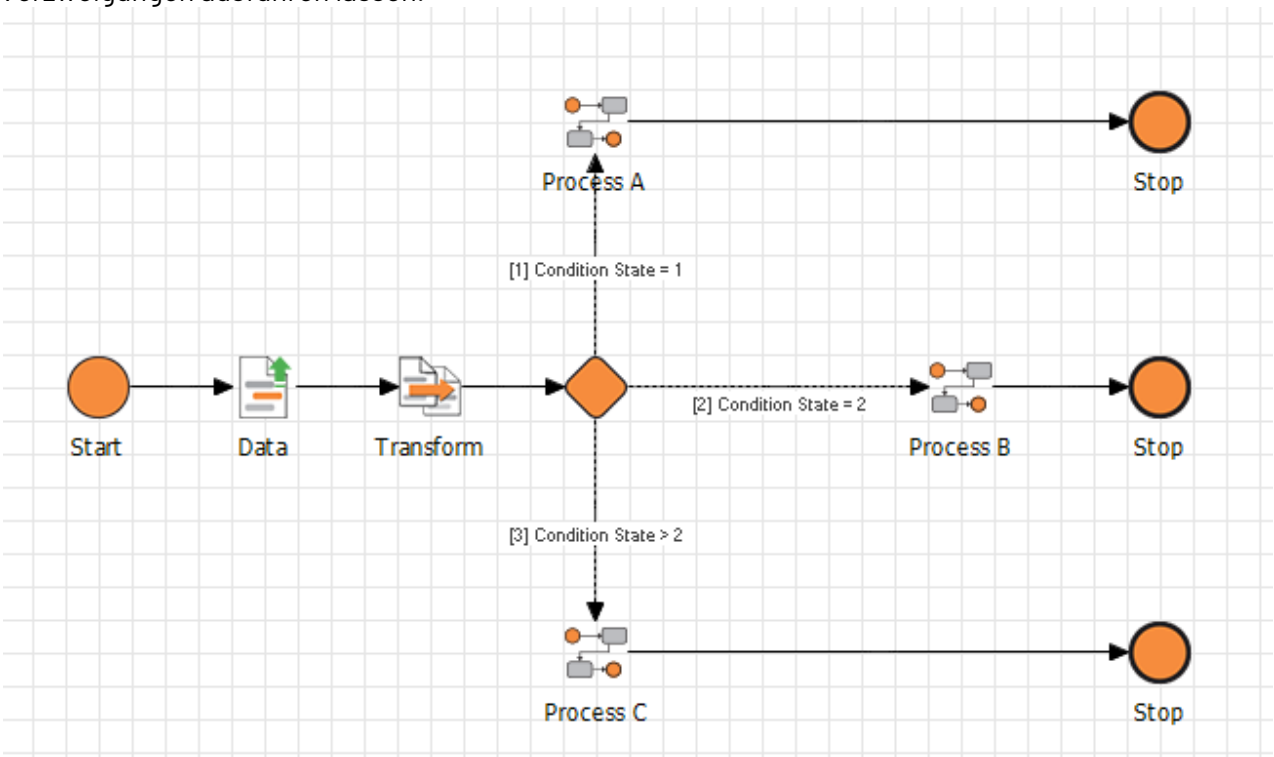
> **Hinweis**
> Ein Prozessbaustein wird mit einem -Symbol gekennzeichnet, sobald für den Prozessbaustein ein **Component-Parameter** festgelegt wird.

*Funktionen in Fenster **ComponentbParameters**:*

- **Add**: Weitere Baustein-Parameter hinzufügen
- **Remove**: Ausgewählten Parameter aus der Liste löschen
- **Up / Down**: Reihenfolge der Parameter wechseln

5.5.5 Status mit ConditionState dynamisch setzen

Mit einem XSL-Mapping können Sie abhängig von verarbeiteten Daten zusätzlich Ausführ-Statuswerte als Processing Instruction erzeugen, die innerhalb des Technical Processes von einem `Condition`-Baustein ausgewertet werden. Je nach Status können Sie so entsprechend definierte Verzweigungen ausführen lassen.



1. Leeres XSL-Mapping in einem Technical Process vor einem `Condition`-Baustein positionieren.
2. Processing Instruction(s) im XSL-Mapping nach folgendem Muster definieren:

```
<xsl:processing-instruction name="ConditionState">value</xsl:processing-
instruction>
```

3. *value* durch einen Wert z.B. aus dem verarbeiteten XML-Dokument ersetzen.
   Beispiel: `<xsl:value-of select="data"/>`

   > **Hinweis**
   > Zulässige Statuswerte *ganze Zahlen (Integer)* sowie negative Zahlen (z. B. *-1*) für einen Fehlerstatus; siehe Statuswerte für Prozesse und Prozess-Schritte
   > Nicht zulässig sind folgende Status-Nummern, da diese systemintern reserviert sind:
   > - *-999* (Fehlerstatus der Engine)
   > - *-998* (Prozess wurde aus Lizenzgründen nicht ausgeführt)
   > - *999* (Status-ID für noch nicht ausgeführte Prozessbausteine)
   > - *998* (wird vom Save Point-Mechanismus verwendet und bewirkt die sofortige Beendigung des gesamten Prozesses)
   > - *997* (bewirkt die sofortige Beendigung des gesamten Technical Processes)
   >
   > **Empfehlung:** Verwenden Sie für benutzerdefinierte Statuswerte Zahlen größer als `1000`, um mögliche Überschneidungen mit anderen Statuswerten zu vermeiden.

4. Sicherstellen, dass das XSL-Mapping ein Wurzel-Element generiert, damit das XSL-Mapping ausgeführt und ein gültiges XML-Dokument erzeugt werden kann.
   Das XSL-Mapping gibt nun *Processing Instructions* nach dem Muster `<?ConditionState value ?>` aus.
5. Im nachfolgenden `Condition`-Baustein für jede Verzweigung einen Status definieren, siehe Bedingungen erstellen und verwenden.
6. Wenn Sie im weiteren Verlauf des Prozesses erneut *ConditionStates* erzeugen: die vorherigen *Processing Instructions* zunächst aus dem Input-XML-Dokument z.B. per `<xsl:copy-of>` entfernen, um unerwartetes Verhalten bei der Prozessausführung zu vermeiden.

> **Problemlösung: Das XSL-Mapping lässt sich nicht ausführen?**
> Möglicherweise ist der verarbeitete Wert einer der unzulässigen Statuswerte *-999*, *-998*, *999*, *998*, *997* oder ist eine Zeichenkette. Oder das vom XSL-Mapping erzeugte XML-Dokument enthält kein Wurzel-Element.
> - Verwenden Sie nur zulässige Statuswerte (siehe oben).
> - Fügen Sie z. B. ein leeres Element `<root/>` nach den *ConditionState*-Definitionen ins XSL-Mapping ein.

5.5.6 Parameter in XSL-Mappings verwenden

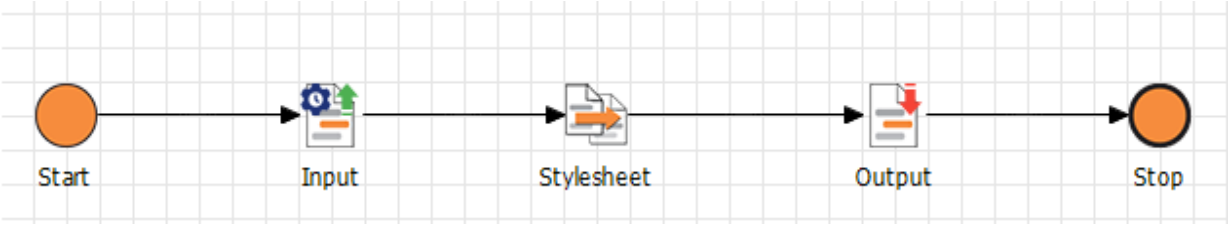
Parameter in XSL-Mappings verwenden

Globale, Projekt- und System-Parameter lassen sich in XSL-Stylesheets verwenden.

Beispiel für die Nutzung

Parameter in XSL-Mappings verwenden

Folgendes Beispiel zeigt die Verwendung von Parametern in einem Mapping. Vor ihrer Verwendung müssen die relevanten Parameter im XSL-Stylesheet eingefügt werden.



Mapping erstellen

Das XSL-Mapping muss nach folgendem Schema aufgebaut sein. Die zu verwendenden Parameter müssen zunächst im XSL-Stylesheet unter ihrem voll qualifizierten Namen deklariert werden. Anschließend lassen sie sich wie gewohnt verwenden.

Beispiel: XSL-Mapping

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
  <xsl:output media-type="text/xml" method="xml"></xsl:output>
  <xsl:param name="PARAM_MyGroup_MyTargetHost"></xsl:param>
  <xsl:param name="PARAM_Thresholds_MinMessages"></xsl:param>
  <xsl:param name="PARAM_Thresholds_MaxMessages"></xsl:param>
  <xsl:template match="/">
    <Result>
      <MyGroup>
        <MyTargetHost><xsl:value-of select="$PARAM_MyGroup_MyTargetHost"></xsl:va
lue-of></MyTargetHost>
      </MyGroup>
      <Thresholds>
        <MinMessages><xsl:value-of select="$PARAM_Thresholds_MinMessages"></xsl:v
alue-of></MinMessages>
        <MaxMessages><xsl:value-of select="$PARAM_Thresholds_MaxMessages"></xsl:v
alue-of></MaxMessages>
      </Thresholds>
    </Result>
  </xsl:template>
</xsl:stylesheet>
```

Technical Process modellieren

1. Neuen Technical Process über das Menü **New > Technical Process** anlegen.
2. Ein Datei für den Input des Mappings in den Prozess ziehen.
3. Das XSL-Mapping (siehe Beispiel) in den Technical Process ziehen.

4. Optional: Eine Datei anlegen, in der der Output des Mappings gespeichert wird.
5. Technical Process speichern und anschließend ausführen.

5.6 Parameter erstellen und verwenden

Wann ist der Einsatz von Parametern sinnvoll?

Wenn Technical Processes für variable Umgebungsbedingungen gerüstet sein sollen, können Parameter die Prozessmodellierung deutlich vereinfachen.

Was sind mögliche Anwendungsfälle?

Mit Parametern können beliebige Schlüssel oder Wertepaare definiert werden: z. B. technische Umgebungsvariablen, wie Pfade, oder fachliche Parameter, wie Grenzwerte für Prüfungen. Zur Laufzeit des Technical Processes werden diese Parameter durch die zuvor definierten Werte ersetzt. Damit lassen sich Technical Processes so modellieren, dass sie in verschiedenen Umgebungsbedingungen ausgeführt werden können, etwa für den Betrieb in Staging-Systemen. Ohne weitere Anpassung der Prozessmodelle lassen sich Technical Processes so über externe Parameter steuern.

Lesen Sie mehr zur Erstellung und Verwendung von Parametern in den folgenden Kapiteln.

Lesen Sie mehr zur Erstellung und Verwendung von Parametern in den folgenden Kapiteln.

- System-Parameter
  - Übersicht der System-Parameter
  - System-Parameter verwenden
- Globale Parameter und -Gruppen
  - Globale Parameter und -Gruppen erstellen
  - Globale Parameter in Projekt-Parameter kopieren
  - Globale Parameter verwenden
- Projekt-Parameter
  - Projekt-Parameter erstellen
  - Projekt-Parameter umbenennen und verschieben
  - Custom Types erstellen
  - Globale und System-Parameter wiederverwenden

5.6.1 System-Parameter

Wie kann ich System-Parameter verwenden?

Die X4 BPMS stellt eine Reihe von System-Parametern bereit, die Sie in Prozessbausteinen einsetzen können. Während der Ausführung des Technical Processes werden die Parameter durch die jeweiligen Werte ersetzt und von der Prozess-Engine verarbeitet.

Weitere Informationen:

- Übersicht der System-Parameter
- System-Parameter verwenden

5.6.1.1 Übersicht der System-Parameter

Folgende Parameter werden in der X4 BPMS bereitgestellt:

`%ACTION_ID%`	Gibt die laufende Nummer des aktuell ausgeführten Prozess-Schrittes aus. *Beispiel:* `3`
`%CALLING_PROCESS%`	Gibt den relativen Repository-Pfad und den Dateinamen des Technical Processes aus, der den aktuellen Technical Process aufgerufen hat. Dieser Parameter wird nur umgewandelt, wenn der aktuelle Technical Process von einem anderen Technical Process direkt als Subprozess aufgerufen wurde. *Beispiel:* `Project/Processes/ProcessCaller.wrf`
`%CALLING_PROCESS_NAME%`	Gibt den Dateinamen (ohne Dateinamenerweitung `*.wrf`) des Technical Processes aus, der den aktuellen Technical Process direkt als Subprozess aufgerufen hat. Dieser Parameter wird nur umgewandelt, wenn der aktuelle Technical Process von einem anderen Technical Process als Subprozess aufgerufen wurde. *Beispiel:* `ProcessCaller`
`%CALLING_PROCESS_PATH%`	Gibt den relativen Repository-Pfad des Technical Processes aus, der den aktuellen Technical Process direkt als Subprozess aufgerufen hat. *Beispiel:* `Project/Processes/`
`%CURRENT_PROCESS%`	Gibt den relativen Repository-Pfad und den Dateinamen des aktuellen Technical Processes aus. *Beispiel:* `Project/Processes/Process.wrf`

%CURRENT_PROCESS_NAME%	Gibt den Dateinamen des aktuellen Technical Processes ohne Dateinamenerweitung (*.wrf) aus. *Beispiel: Process*
%CURRENT_PROCESS_PATH%	Gibt den relativen Repository-Pfad des aktuellen Technical Processes aus. Dieser Parameter wird nur umgewandelt, wenn der aktuelle Technical Process von einem anderen Technical Process als Subprozess aufgerufen wurde. *Beispiel: Project/Processes/*
%CURRENT_PROJECT%	Gibt den Namen des aktuellen Repository-Projekts aus. *Beispiel: Project*
%HOSTNAME%	Gibt den Host-Namen des *X4 Servers* aus. *Beispiel: YC-WS-000.yourcompany.local*
%HTTP_PORT%	Gibt den HTTP-Port aus, der in der Konfigurationsdatei x4config.xml im Element <webContainerURL/> festgelegt wurde. *Beispiel: 8080*
%INSTANCE%	Gibt den Namen der aktuellen *X4 Server*-Instanz aus, der in der Konfigurationsdatei x4config.xml im Element <instanceName> festgelegt wurde. *Beispiel: X4*
%IPADRESS%	Gibt die IP-Adresse des *X4 Servers* aus. *Beispiel: 192.168.0.1*
%NOW%	Gibt das aktuelle Datum und die Uhrzeit im ISO-Format yyyy-MM-ddThh:mm:ss aus. *Beispiel: 2021-08-11T17:46:50*
%NOW_PLAIN%	Gibt das aktuelle Datum und die Uhrzeit im Format yyyyMMddhhmmss aus. *Beispiel: 20210811174650*

`%PID%`	Gibt die ID der aktuellen Prozess-Instanz als ganze Zahl aus, die innerhalb einer *X4 Server*-Instanz eindeutig ist. Subprozesse werden mit derselben Prozess-ID ausgeführt, nicht jedoch Technical Processes, die über den *Process Starter* gestartet wurden. *Beispiel:* `1628696804892237809` Wenn mehrere Instanzen eines *X4 Servers* parallel auf einem Applikations-Server ausgeführt werden, empfiehlt sich, statt `%PID%` den eindeutigen Platzhalter `%UID%` zu verwenden.
`%PUID%`	Gibt die eindeutige UUID des Start-Prozesses aus.
`%START_PROCESS%`	Gibt den relativen Repository-Pfad und den Dateinamen des Technical Processes aus, der ursprünglich gestartet wurde. Wird der aktuelle Technical Process als Subprozess eines anderen Technical Processes gestartet, so wird dessen Pfad und Dateiname angezeigt. *Beispiel:* `Project/Processes/ProcessCaller.wrf`
`%START_PROCESS_NAME%`	Gibt den Dateinamen des ursprünglich gestarteten Technical Processes ohne Dateinamenerweierung (`*.wrf`) aus. Wird der aktuelle Technical Process als Subprozess gestartet, so wird der Dateiname des ursprünglichen gestarteten Technical Processes angezeigt, der den aktuellen Prozess-Kontext erzeugt hat. ⓘ Beachten Sie, dass bei einem Prozessaufruf mittels *Process Starter* ein neuer Prozess-Kontext erzeugt wird. *Beispiel:* `ProcessCaller`
`%START_PROCESS_PATH%`	Gibt den relativen Repository-Pfad des Technical Processes aus, der ursprünglich gestartet wurde. Wird der aktuelle Technical Process als Subprozess eines anderen Technical Processes gestartet, so wird dessen Pfad angezeigt. *Beispiel:* `Project/Processes/`
`%TIME_MS%`	Gibt die Zahl der Millisekunden seit 1. Januar 1970, 0:00 Uhr aus. *Beispiel:* `1628696810712`

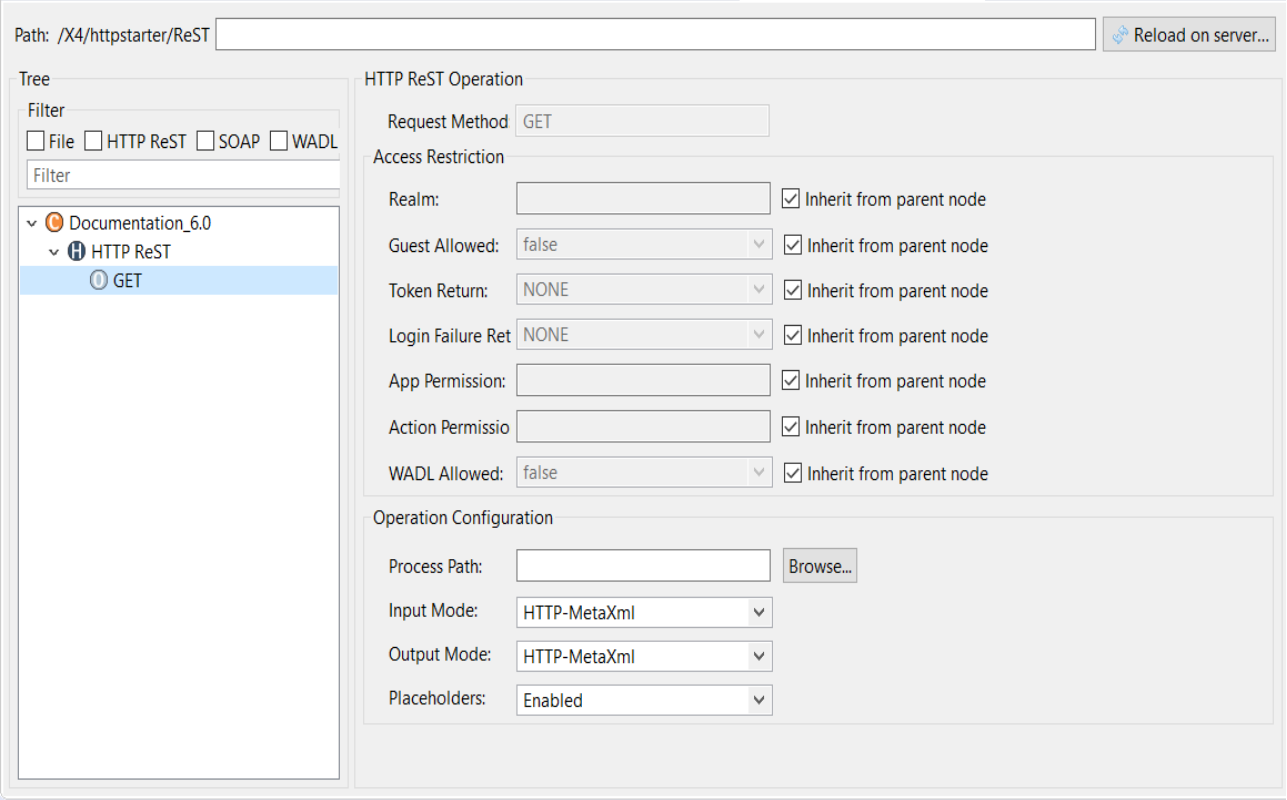
%UID%	Gibt den Namen der *X4 Server*-Instanz %INSTANCE% und die angehängte aktuelle Prozess-ID (%PID%) aus. Damit lässt sich eine Prozess-Instanz eindeutig identifizieren, wenn auf demselben Applikations-Server mehrere *X4 Server*-Instanzen parallel ausgeführt werden. *Beispiel: X41628696804892237809*
%USER_COLLECTION%	Gibt den Pfad (absolut oder relativ) zum Repository-Ordner des aktuellen Benutzers in der X4DB aus; die so genannte *"User Collection"*. Dieser Wert wird aus der zentralen Konfigurationsdatei X4config.xml von Element <repositoryPath> unverändert übernommen, der dort auch relativ definiert sein kann, siehe *X4 Server - Repository-Konfiguration* *Beispiel: C:\X4\X4DB\1* oder *./X4DB/1*
%USER_COLLECTION_ABSOLUTE%	Gibt den absoluten Pfad zum Repository-Ordner des aktuellen Benutzers in der X4DB aus; die so genannte *"User Collection"* *Beispiel: C:\X4\Server\X4DB\1*
%USER_ID%	Gibt die ID des aktuellen Benutzers aus, in dessen Repository-Ordner sich der Technical Process befindet (*"user collection"*). Seit X4-BPMS-Version 4.0 ist die User-ID keine ganze Zahl mehr, sondern eine Zeichenkette. *Beispiel:1*
%WORKING_DIR_ABSOLUTE%	Gibt den absoluten Pfad des aktuellen Arbeitsverzeichnisses aus, in dem der Applikations-Server gestartet wurde. *Beispiel: C:\X4\Server*

Parameter für Webservices

Innerhalb des Webservice Configuration Editors können zusätzliche Parameter für die Nutzung von Webservices aktiviert werden. Damit lassen sich Parameter für diese ReSTful-Webservice- oder SOAP-Webservice-Operation verarbeiten und Informationen aus dem HTTP-Request als Parameter bereitstellen (als effizientere Alternative zu Input Mode HTTP-MetaXml).

Mögliche Werte:

- Disabled: Keine Parameter aus dem HTTP-Request im Kontext des Technical Processes bereitstellen (Standard)
- Enabled: Parameter aus HTTP-Request-Informationen im Kontext des Technical Processes bereitstellen



Folgende Parameter mit Informationen aus dem HTTP-Request stehen bei Option `Enabled` innerhalb des X4-Prozesskontexts zur Verfügung, z.B. innerhalb von XSL-Mappings:

- `REQUEST_PARAMETER_<`*Name*`>`: Gewünschter Parameter (Groß-/Kleinschreibung wird beachtet!)
- `REQUEST_HEADER_<`*NAME*`>`: Gewünschter HTTP-Header (nur Großschreibung!)
- `REQUEST_METHOD`
- `REQUEST_SCHEME`
- `REQUEST_PATH`
- `REQUEST_FULL_PATH`
- `REQUEST_REMOTE_ADDRESS`
- `REQUEST_REMOTE_PORT`
- `REQUEST_REMOTE_USER`
- `REQUEST_LOCAL_ADDRESS`
- `REQUEST_LOCAL_PORT`
- `REQUEST_SERVER_NAME`
- `REQUEST_SERVER_PORT`

ⓘ Diese Parameter können wie alle Standard-Platzhalter verwendet werden, lassen sich jedoch nicht überschreiben und sind nicht in der Verwaltung für globale Parameter sichtbar.

5.6.1.2 System-Parameter verwenden

Für System-Parameter sind folgende Verwendungen möglich:

System-Parameter in Adaptern verwenden

Wie globale und Projekt-Parameter können auch System-Parameter in Adapterkonfigurationen verwendet werden (siehe Anwendungsbeispiele für Parameter).

System-Parameter in XSL-Mappings verwenden

System-Parameter lassen sich als Variablen mit `$Parameter` in XSL-Mappings verwenden, zum Beispiel `$PID`.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output media-type="text/xml" method="xml"/>
<xsl:param name="PID"/>
<xsl:param name="NOW"/>
<xsl:template match="/">
   <user>
      <ID>
         <xsl:value-of select="$PID"/>
      </ID>
      <Date>
         <xsl:value-of select="$NOW"/>
      </Date>
   </user>
</xsl:template>
</xsl:stylesheet>
```

5.6.2 Globale Parameter und -Gruppen

Wie kann ich globale Parameter verwenden?

Globale Parameter lassen sich manuell über die **Global Parameter Configuration** definieren. Dabei können beliebige Parametergruppen mit globalen Parametern erstellt und direkt im X4 Designer verwaltet werden. Diese lassen sich anschließend wie System-Parameter und Projekt-Parameter in den Werkzeugen des X4 Designers (Adapter Editor, Mapping Editor, Process Editor) nutzen.

Weitere Informationen:

- Globale Parameter und -Gruppen erstellen
- Globale Parameter in Projekt-Parameter kopieren
- Globale Parameter verwenden

5.6.2.1 Parameter in der Datenbank persistieren

Es steht ein SQL Parameter Storage-Plugin zu Verfügung zum Persistieren von Parametern in der Datenbank (siehe SQL Parameter Storage-Plugin).

5.6.2.2 Parameter über den Global Parameter Management verwalten

Mit dem Adapter Global Parameter Management lassen sich globale Parameter anlegen und verwalten. Dabei bietet der Adapter verschiedene Operationen zur Verwaltung globaler Parameter, etwa zum Hinzufügen oder Löschen, aber auch zum Import und Export von Parametern oder Parametergruppen.

5.6.2.3 Parameter im X4 Control Center anpassen

Während des Betriebs können Fachadministratoren im X4 Control Center über Administration > X4 API (Swagger UI) die globale Parameterkonfiguration jederzeit einsehen und anpassen. Neue Prozess-Instanzen werden anschließend mit geänderten Parametern ausgeführt.

5.6.2.4 Globale Parameter und -Gruppen erstellen

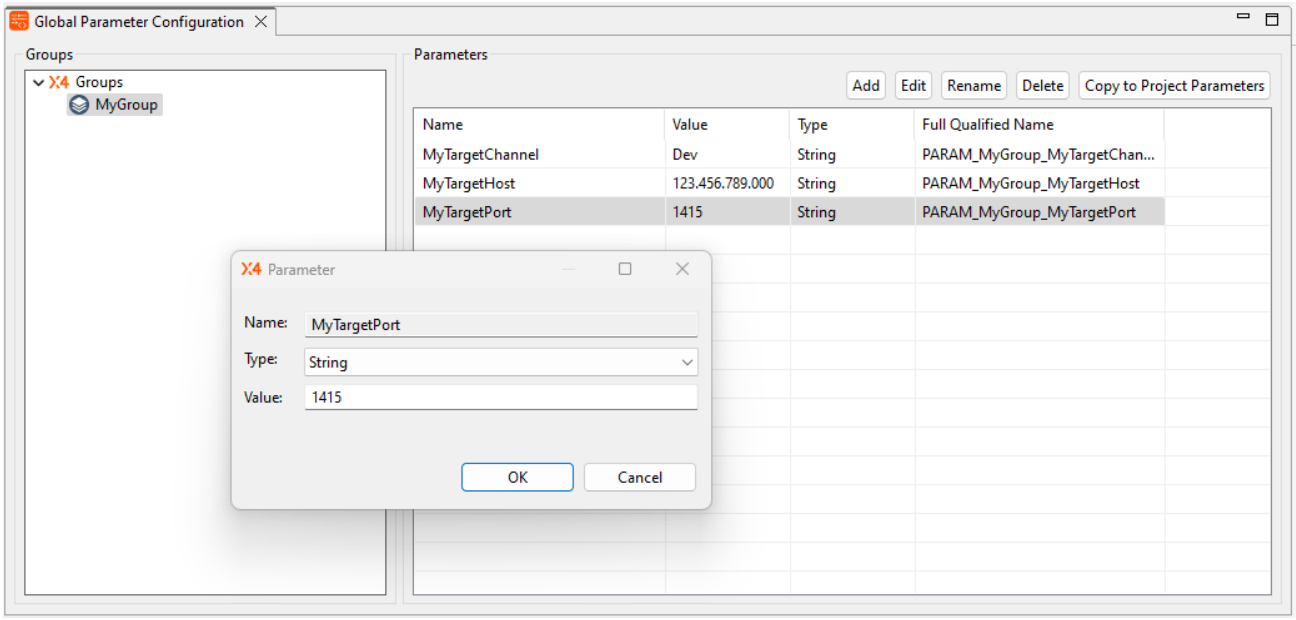
Globale Parameter und -gruppen können in der **Global Parameter Configuration** erstellt und verwaltet werden.

Hinweis
Beachten Sie bei der Verwendung von Parametern folgende Priorisierung:
System-Parameter → Projekt-Parameter → globale Parameter
Bei Namenskonflikten wird eine Fehlermeldung bzw. Warnung in der Sicht **Problems** angezeigt:

- Fehlermeldung bei Namenskonflikten von System- und Projekt-Parametern
- Warnung bei Namenskonflikten von Projekt-Parametern und globalen Parametern

So rufen Sie die Global Parameter Configuration auf

Wählen Sie das Menü **Tools > Global Parameter Configuration**:



5.6.2.4.1 Neue Gruppe mit globalen Parametern erstellen

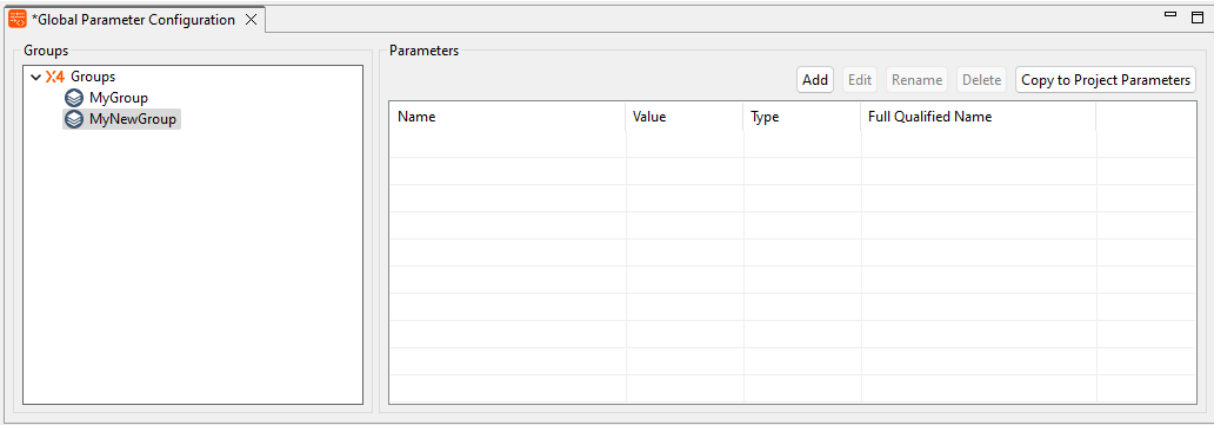
> **Hinweis**
> Um Namenskonflikte zu vermeiden, muss der Name der Parameter-Gruppe eindeutig sein.

So erstellen Sie eine Gruppe mit globalen Parametern

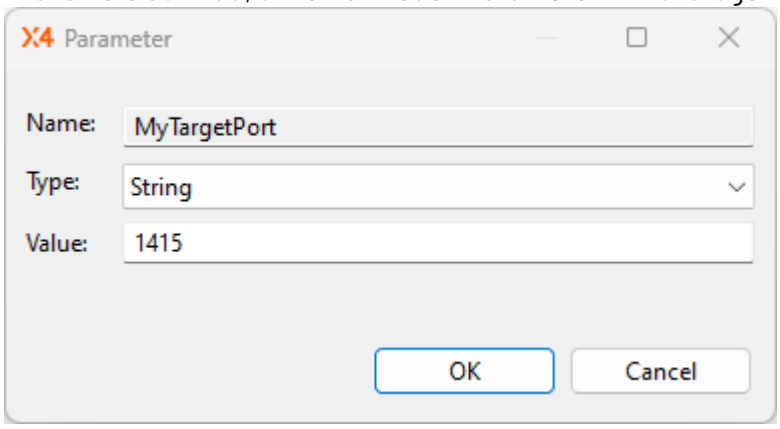
1. Öffnen Sie die **Global Parameter Configuration** über das Menü **Tools > Global Parameter Configuration**.

2. Öffnen Sie das Kontextmenü mit einem Rechtsklick auf **Groups**.
3. Wählen Sie **Add** im Kontextmenü.
4. Geben Sie einen Namen für die Parametergruppe ein und bestätigen Sie mit **OK**.
   Die neue Gruppe wurde hinzugefügt und erscheint als neuer Knoten unterhalb des Wurzelknotens.

5. Wählen Sie eine Gruppe, um neue Parameter hinzuzufügen.



6. Klicken Sie auf **Add**, um einen neuen Parameter hinzuzufügen.



7. Geben Sie die Eigenschaften für den Parameter ein:

Datenfeld	Beschreibung
Name	Name des Parameters **Hinweis** Um Namenskonflikte zu vermeiden, muss der Name der Parameter-Gruppe eindeutig sein. **Hinweis** Folgende Zeichen sind hierbei erlaubt: `^[a-zA-Z][a-zA-Z0-9.\-]*$` Ein Punkt im Namen sollte vermieden werden, wenn der globale Parameter in die **Project Parameter Configuration** kopiert werden soll. **Beispiel:** `My-Global-Parameter-123`

Datenfeld	Beschreibung
Type	Datentyp des Parameters Mögliche Werte: • String • Boolean • Integer • Decimal • Date • Password **Hinweis** Für Parameter des Typs `Password` lässt sich zusätzlich die Option **Show Password** aktivieren, um das Passwort unverschlüsselt anzuzeigen.
Value	Wert des Parameters

8. Klicken Sie auf **OK**, um den Parameter hinzuzufügen.
   Der Parameter wurde angelegt. Der vollqualifizierte Name des Parameters wird dabei automatisch angelegt.

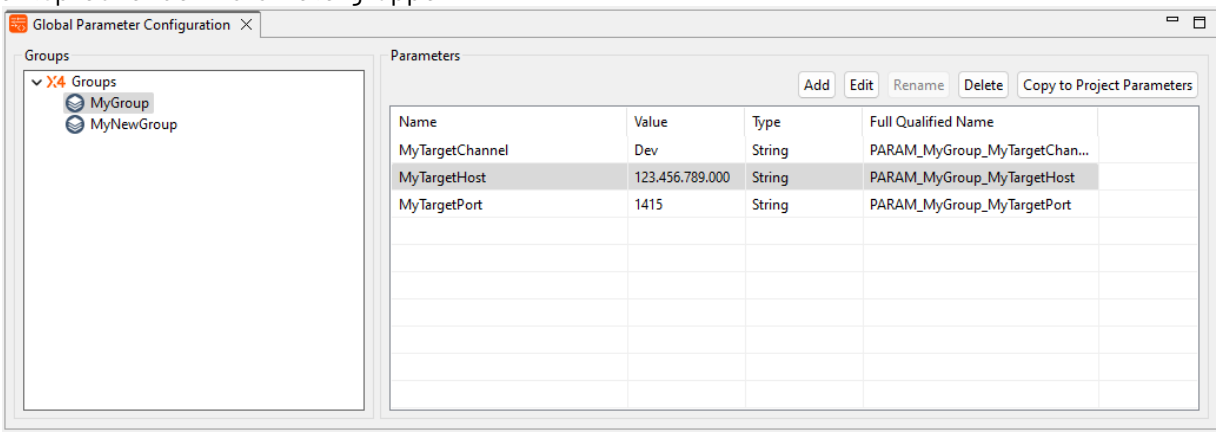
Hinweis
Über Rechtsklick auf einen globalen Parameter lässt sich dessen Kontextmenü öffnen. Mit **Copy Full Qualified Name to Clipboard** kann anschließend dessen vollqualifizierter Name kopiert und an anderer Stelle eingefügt werden.

5.6.2.4.2 Globale Parameter umbenennen

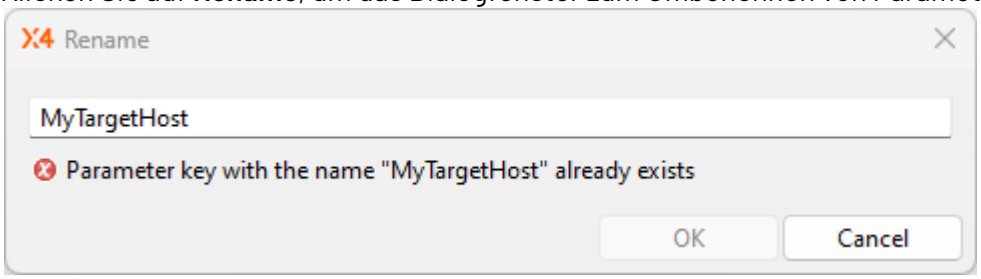
Hinweis
Die Aktion **Rename** ist nur aktiv, wenn alle Änderungen in **Global Parameter Configuration** gespeichert sind.

So benennen Sie einen einzelnen Parameter um und aktualisieren dessen Referenzen dabei automatisch

1. Wählen Sie in **Global Parameter Configuration** den gewünschten Parameter in der entsprechenden Parametergruppe.



2. Klicken Sie auf **Rename**, um das Dialogfenster zum Umbenennen von Parametern zu öffnen:

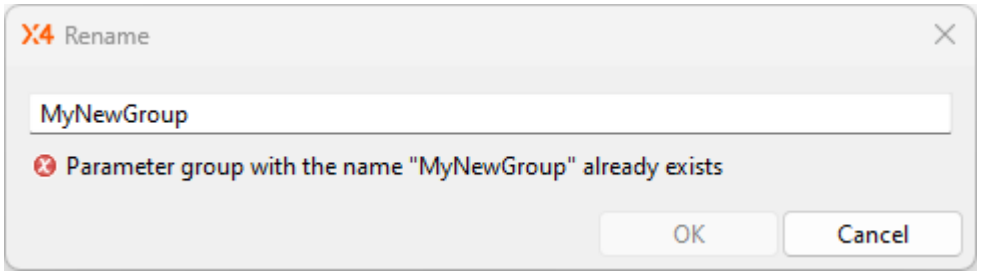


3. Geben Sie den neuen Namen für den Parameter ein.
4. Klicken Sie auf **OK**, um die gewünschten Änderungen durchzuführen.

5.6.2.4.3 Gruppe mit globalen Parametern umbenennen

So benennen Sie eine Gruppe mit globalen Parametern um und aktualisieren deren Referenzen dabei automatisch

1. In **Global Parameter Configuration**, öffnen Sie das Kontextmenü mit einem Rechtsklick auf die gewünschte Parametergruppe.
2. Klicken Sie auf **Rename Group**, um das Dialogfenster zum Umbenennen von Parametergruppen zu öffnen:



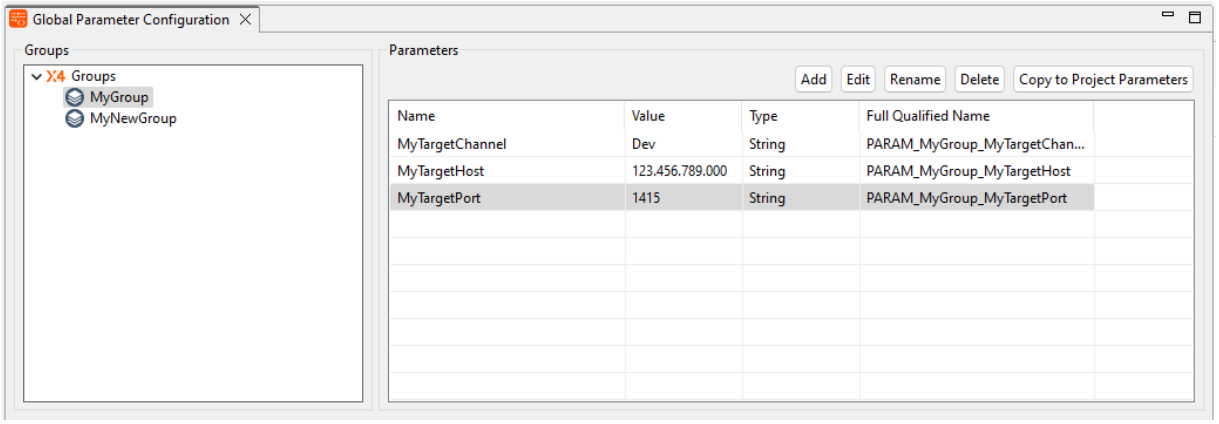
3. Geben Sie einen neuen Namen ein.
4. Klicken Sie auf **OK**, um die gewünschten Änderungen durchzuführen.

5.6.2.4.4 Globale Parameter bearbeiten

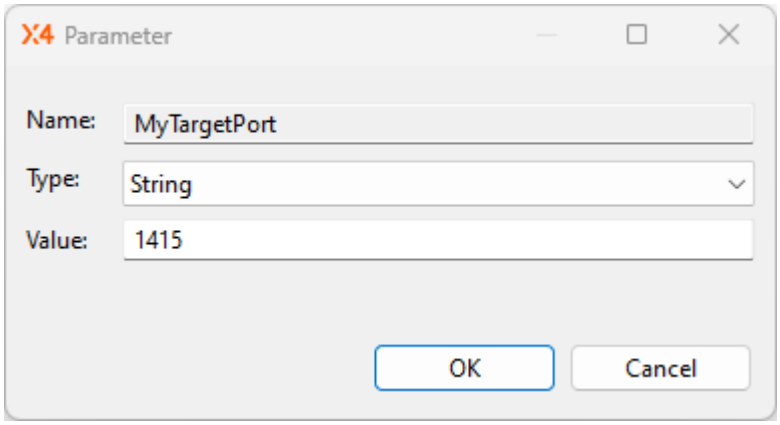
> **Hinweis**
> Durch Bearbeiten können Sie den Typ (**Type**) und Wert (**Value**) eines vorhandenen globalen Parameters ändern.

So editieren Sie einen einzelnen globalen Parameter

1. Wählen Sie in **Global Parameter Configuration** den gewünschten Parameter in der entsprechenden Parametergruppe.



2. Klicken Sie **Edit** oder öffnen Sie per Doppelklick das Dialogfenster zum Editieren des Parameters:



3. Editieren Sie den Parameter.
4. Klicken Sie auf **OK**, um die gewünschten Änderungen durchzuführen.

5.6.2.5 Globale Parameter in Projekt-Parameter kopieren

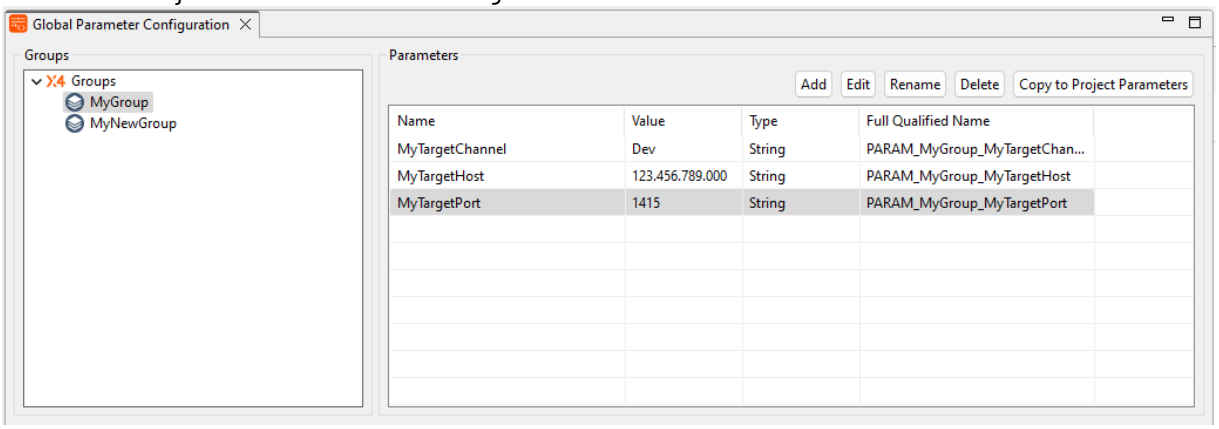
Globale Parameter können zu den Projekt-Parametern hinzugefügt werden. Globale Parameter aus Gruppen werden in der **Project Parameter Configuration** auf Root-Level-Ebene eingefügt, d.h. ohne die Gruppenzuordnung aus der **Global Parameter Configuration**.

> **Hinweis**
> Beachten Sie bei der Verwendung von Parametern folgende Priorisierung:
> System-Parameter → Projekt-Parameter → globale Parameter
> Bei Namenskonflikten wird eine Fehlermeldung bzw. Warnung in der Sicht **Problems** angezeigt:
>
> - Fehlermeldung bei Namenskonflikten von System- und Projekt-Parametern
> - Warnung bei Namenskonflikten von Projekt-Parametern und globalen Parametern

So fügen Sie globale Parameter zu Projekt-Parametern hinzu

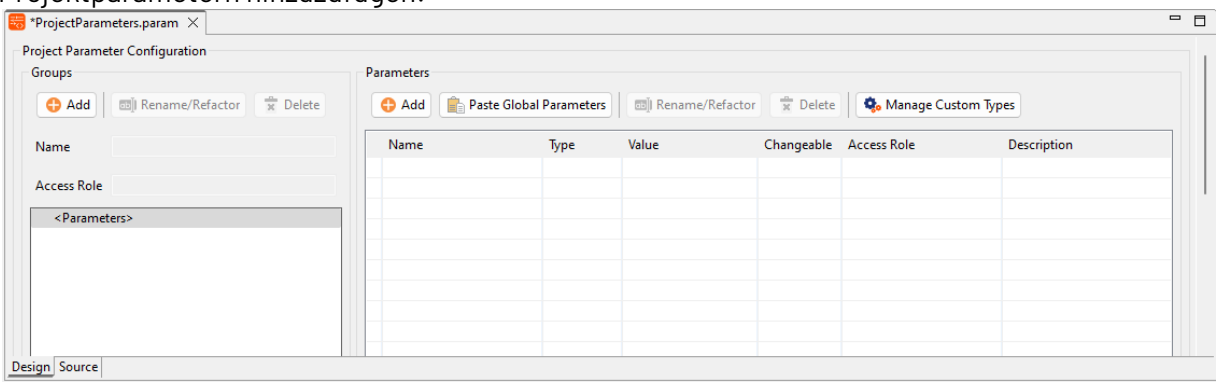
> **Hinweis**
> Löschen oder Umbenennen von Parametern nach der Aktion **Copy to Project Parameters**
> Wenn ein globaler Parameter zu den Projekt-Parametern kopiert wird, existieren zwei Parameter mit demselben Namen. Hierzu erscheint in der Sicht **Problems** eine Warnung. Dennoch kann dies z.B. bei der Auswahl der Parameter in **Mappings**, **Conditions** und **Adaptern** verwirrend sein. Daher wird empfohlen, den globalen Parameter zu löschen oder den Projekt-Parameter umzubenennen.

1. Wählen Sie in **Global Parameter Configuration** einen oder mehrere globale Parameter aus, die Sie zu den Projekt-Parametern hinzufügen möchten.



2. Klicken Sie auf `Copy to Project Parameters`.
3. Öffnen Sie die Datei mit der Endung `*.param` (**Project Parameter Configuration**) in dem Projekt, zu dem die globalen Parameter hinzugefügt werden sollen.

4. Klicken Sie auf `Paste Global Parameters`, um die globalen Parameter zu den Projektparametern hinzuzufügen:



Die globalen Parameter wurden nun der **Project Parameter Configuration** hinzugefügt.

5.6.2.6 Globale Parameter verwenden

> **Wie kann ich globale Parameter verwenden?**
> Globale Parameter lassen sich über die **Global Parameter Configuration** definieren. Dabei können beliebige Parametergruppen mit globalen Parametern erstellt und direkt im X4 Designer verwaltet werden.
> Diese lassen sich anschließend wie System-Parameter und Projekt-Parameter in den Werkzeugen des X4 Designers (Adapter Editor, Mapping Editor, Process Editor) nutzen.

Globale Parameter in Adaptern verwenden

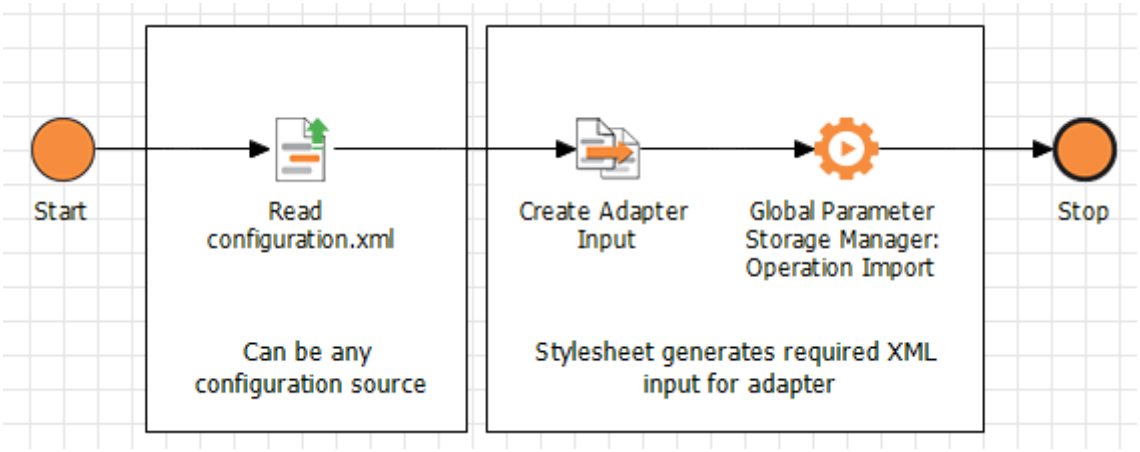
Analog zu System- und Projekt-Parametern lassen sich auch globale Parameter mit `%PARAM_Gruppe_Name%` in Adapter-Konfigurationen einsetzen (siehe Anwendungsbeispiele für Parameter).

Globale Parameter in Technical Processes verwenden

Um globale Parameter in Technical Processes verwenden zu können, müssen die zu verwendenden Parameter zunächst in den aktuellen Prozess-Kontext eingelesen werden. Mit dem Adapter Global Parameter Management lassen sich globale Parameter über X4-Prozesse anlegen und zur Prozesslaufzeit verwalten. Dabei bietet der Adapter verschiedene Operationen zur Verwaltung globaler Parameter, etwa zum Hinzufügen oder Löschen, aber auch zum Import und Export von Parametern oder Parametergruppen.

Globale Parameter mit dem Global Parameter Management Adapter anlegen

Folgendes Prozessbeispiel zeigt, wie globale Parameter mit einem Mapping über den Global Parameter Management Adapter angelegt werden können. Der Technical Process liest dabei ein XML-Dokument mit Konfigurationswerten ein, transformiert es über ein Stylesheet und erstellt die Parameter über den Adapter mit der passenden Operation. Somit sind die Parameter für das Projekt und dessen Prozesse verfügbar.



Input-XML-Dokument bereitstellen

Das Input-XML-Dokument muss nach folgendem Schema aufgebaut sein:

Beispiel: Input-XML mit vordefinierten Parametern

```
<Placeholder>
    <Group name="MyGroup">
        <Key name="MyTargetHost" type="String">123.456.789.000</Key>
        <Key name="MyTargetPort" type="Integer">1415</Key>
        <Key name="MyTargetChannel" type="String">Dev</Key>
    </Group>
    <Group name="Thresholds">
        <Key name="MinMessages" type="Integer">0</Key>
        <Key name="MaxMessages" type="Integer">10</Key>
    </Group>
</Placeholder>
```

Erläuterung:

- Das Wurzelelement `Placeholder` kann beliebig viele `Group`-Elemente enthalten.
- Ein `Group`-Element stellt jeweils eine cdar und kann beliebig viele `Key`-Elemente enthalten.

- Das Attribute `name` innerhalb eines `Group`-Elements definiert den Namen der Parametergruppe.
- Ein `Key` steht jeweils für einen Parameter und enthält folgende Attribute und Inhalte:
  - `name`: Name des Parameters
  - `type`: Parametertyp `String`, `Boolean`, `Integer`, `Decimal` oder `Date` (ISO-Date-Format `YYYY-MM-DD`)
  - Wert des Parameters entsprechend dem Parametertyp

So modellieren Sie den Technical Process

1. Legen Sie einen neuen Technical Process über das Menü **New > Technical Process** an.
2. Ziehen Sie die Input-XML-Datei (siehe Beispiel) in den Technical Process und wählen Sie in den Eigenschaften die Operation *`Read`*.
3. Fügen Sie ein Stylesheet ein zur Transformation der Konfigurationsdatei (Inputdatei) auf die Input-Struktur, die der nachfolgende Adapter benötigt.
4. Ziehen Sie den Global Parameter Management Adapter in den Technical Process und wählen Sie die Adapter-Operation *`AddKeys`*.
5. Speichern Sie den Technical Process.
6. Führen Sie den Technical Process aus.
   Die Parameter wurden nun angelegt und können sowohl in der Global Parameter Configuration als auch in der Administration-Oberfläche des X4 Control Centers eingesehen und bearbeitet werden.

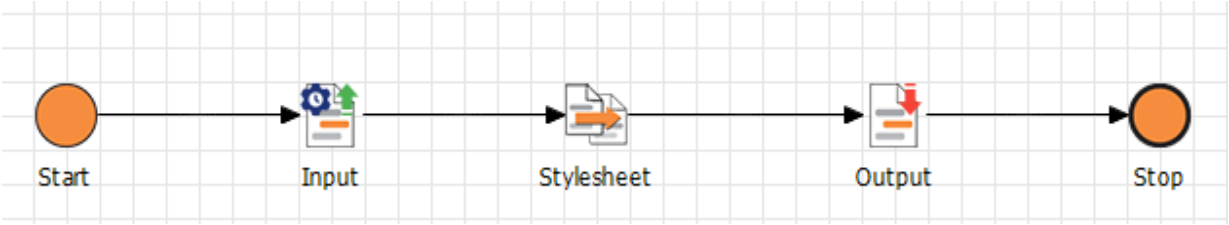
Globale Parameter in XSL-Mappings verwenden

Analog zu den Projekt- und System-Platzhaltern lassen sich auch globale Parameter mit `%PARAM_Group_Name%` in XSL-Stylesheets verwenden.

Beispiel für die Nutzung

Globale Parameter in XSL-Mappings verwenden

Folgendes Beispiel zeigt die Verwendung von globalen Parametern in Mappings. Vor ihrer Verwendung müssen die relevante Parameter im XSL-Mapping eingefügt werden.



Mapping erstellen

Das XSL-Mapping muss nach folgendem Schema aufgebaut sein. Die zu verwendenden globalen Parameter müssen zunächst im Stylesheet unter ihrem voll qualifizierten Namen deklariert werden. Anschließend lassen sie sich wie gewohnt verwenden.

Beispiel: XSL-Mapping

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
  <xsl:output media-type="text/xml" method="xml"></xsl:output>
  <xsl:param name="PARAM_MyGroup_MyTargetHost"></xsl:param>
  <xsl:param name="PARAM_Thresholds_MinMessages"></xsl:param>
  <xsl:param name="PARAM_Thresholds_MaxMessages"></xsl:param>
  <xsl:template match="/">
    <Result>
      <MyGroup>
        <MyTargetHost><xsl:value-of select="$PARAM_MyGroup_MyTargetHost"></xsl:va
lue-of></MyTargetHost>
      </MyGroup>
      <Thresholds>
        <MinMessages><xsl:value-of select="$PARAM_Thresholds_MinMessages"></xsl:v
alue-of></MinMessages>
        <MaxMessages><xsl:value-of select="$PARAM_Thresholds_MaxMessages"></xsl:v
alue-of></MaxMessages>
      </Thresholds>
    </Result>
  </xsl:template>
</xsl:stylesheet>
```

Technical Process modellieren

1. Neuen Technical Process über das Menü **New > Technical Process** anlegen.
2. Das XSL-Mapping (siehe Beispiel) in den Technical Process ziehen.
3. Technical Process speichern und anschließend ausführen.

5.6.3 Projekt-Parameter

Wie kann ich Projekt-Parameter verwenden?

In einem bestehenden Projekt können Sie die **Project Parameter Configuration**, eine Datei mit der Endung `*.param`, mit projektspezifischen Parametern und Parameter-Gruppen anlegen.

Zudem können Sie dort **Custom Types** (Enums) mit **Custom Type Values** (Enum-Werten) deklarieren, die Sie später beim Erstellen eines Projekt-Parameters auswählen können. Ein möglicher Anwendungsfall von **Custom Types** mit **Custom Type Values** sind Bedingungen mit `Condition`-Baust einen (siehe Bedingungen erstellen und verwenden).

Projekt-Parameter lassen sich anschließend wie globale und System-Parameter in den Werkzeugen des X4 Designers (Adapter Editor, Mapping Editor, Process Editor, Condition Editor, Properties View) nutzen und können mit dem Projekt exportiert bzw. importiert werden.

Projekt-Parameter und Projekt-Parameter-Gruppen werden beim Start des Servers geladen und mit dem jeweiligen Projekt deployed.

Sie können zur Laufzeit über ReST API, X4 Adapter oder Web Apps geändert und persistiert werden.

Vorteile von Projekt-Parametern

- Wiederverwendbarkeit von Komponenten durch Parametrierung wird erhöht.
- Der Prozess kann durch Projekt-Parameter direkt zur Laufzeit beeinflusst werden.
- Projekt-Parameter können mit der `*.param`-Datei direkt mit dem Projekt exportiert oder importiert werden.
- Projekt-Parameter können zur Laufzeit über das X4 Control Center (Swagger UI) oder über einen speziellen Adapter, den Adapter Project Parameter Management, geändert werden.
- Projektparameter können, genau wie System- und globale Parameter, in Mappings, Conditions und weiteren Konfigurationen, wie z.B. Adapter, sowie in Business Rules oder Human Tasks (BPM-Projekten) verwendet werden.
- Globale Parameter können in die Project Parameter Configuration kopiert werden, um dort projektspezifisch genutzt zu werden, siehe Globale Parameter in Projekt-Parameter kopieren.

Besonderheiten von Projekt-Parametern

- **Scoping**: Ein Parameterwert kann von allen Prozessen gemeinsam genutzt werden oder für jede Prozessausführung unterschiedlich sein.
- **Kennwortverschlüsselung**: Projekt-Parameter vom Typ **Password** werden über einen Laufzeitmechanismus verschlüsselt. Zur Entwicklungszeit sollten keine Kennwörter in der Datei konfiguriert werden.
- **Parameter vom Typ Constant (read only)**: Ein Parameter kann als schreibgeschützt definiert werden, so dass er zur Laufzeit nicht geändert werden kann.
- **Zugriffsbeschränkung**: Sie können festlegen, dass ein Parameter nur von Benutzern mit einer bestimmten Rolle im Authentifizierungsprovider Keycloak bearbeitet werden kann (siehe https://www.keycloak.org im Bereich Docs> Server Administration).
- **Metadaten**: Metadaten (Beschreibung) liefern Informationen zum Parameter für die Konfiguration.

- **X4 Control Center**: Über das X4 Control Center (Swagger UI) können alle Parameter angezeigt werden, die der aktuell angemeldete Benutzer sehen oder bearbeiten darf.

Weitere Informationen:

- Projekt-Parameter erstellen
- Projekt-Parameter umbenennen und verschieben
- Custom Types erstellen
- Globale und System-Parameter wiederverwenden

5.6.3.1 Projekt-Parameter über den Project Parameter Management verwalten

Mit dem Adapter Project Parameter Management lassen sich Projekt-Parameter anlegen und verwalten. Dabei bietet der Adapter verschiedene Operationen zur Verwaltung von Projekt-Parametern, etwa zum Hinzufügen oder Löschen, aber auch zum Import und Export von Parametern oder Parametergruppen.

5.6.3.2 Projekt-Parameter im X4 Control Center steuern

Während des Betriebs können Fachadministratoren im X4 Control Center über Administration > X4 API (Swagger UI) die Konfiguration von Projekt-Parametern jederzeit einsehen und anpassen. Neue Prozess-Instanzen werden anschließend mit geänderten Parametern ausgeführt.

5.6.3.3 Projekt-Parameter erstellen

Projekt-Parameter und Projekt-Parameter-Gruppen können in einer projekt-spezifischen Datei mit der Endung `*.param` (**Project Parameter Configuration**) erstellt werden.

Projekt-Parameter lassen sich anschließend wie globale und System-Parameter in den Werkzeugen des X4 Designers (Adapter Editor, Mapping Editor, Process Editor, Condition Editor, Properties View) nutzen und können mit dem Projekt exportiert bzw. importiert werden.

Projekt-Parameter und Projekt-Parameter-Gruppen werden beim Start des Servers geladen und mit dem jeweiligen Projekt deployed.

Sie können zur Laufzeit über ReST API, X4 Adapter oder Web Apps geändert und persistiert werden.

Hinweis
Beachten Sie bei der Verwendung von Parametern folgende Priorisierung:
System-Parameter → Projekt-Parameter → globale Parameter
Bei Namenskonflikten wird eine Fehlermeldung bzw. Warnung in der Sicht **Problems** angezeigt:

- Fehlermeldung bei Namenskonflikten von System- und Projekt-Parametern
- Warnung bei Namenskonflikten von Projekt-Parametern und globalen Parametern

Hinweis
Mit der Tastenkombination `Strg+Leertaste` können in Projektdateien (Prozessbausteinen, Mappings, Adaptern und Conditions) alle verfügbaren Parameter angezeigt und ausgewählt werden.

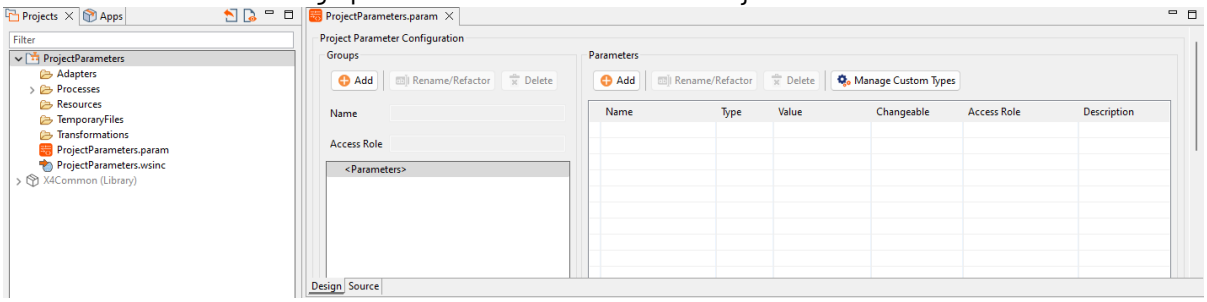
Auf dieser Seite finden Sie folgende Inhalte:

- Project Parameter Configuration erstellen
- Projekt-Parameter hinzufügen
- Projekt-Parameter-Gruppen hinzufügen
- Beispiel einer Project Parameter Configuration
  - Project Parameter Configuration mit Constant und Variable auf Root Level
  - Project Parameter Configuration mit Projekt-Parameter-Gruppe
  - Beispiel der XML-Struktur

5.6.3.3.1 Project Parameter Configuration erstellen

So erstellen Sie die Project Parameter Configuration

1. Wählen Sie ein bestehendes ESB-Projekt und klicken Sie mit der rechten Maustaste auf den Projektordner.
2. Wählen Sie im Kontextmenü **New** > **Project Parameter Configuration**
   Eine Datei mit der Endung `.param` und dem Namen des Projekts wird erstellt:

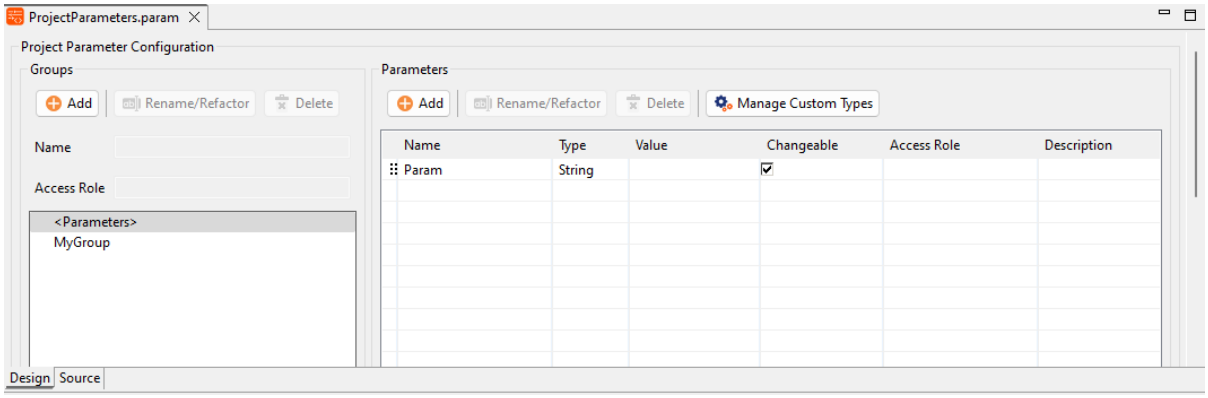


5.6.3.3.2 Projekt-Parameter hinzufügen

> **Hinweis**
>
> - Das Bearbeiten von Projekt-Parametern und Projekt-Parameter-Gruppen kann durch Rollen im Authentifizierungsprovider Keycloak eingeschränkt werden (siehe https://www.keycloak.org im Bereich **Docs> Server Administration**).
> - Parameter können über die **Design**-Ansicht oder über die **Source**-Ansicht direkt in der XML-Struktur hinzugefügt werden.
> - Parameter vom Typ Enum können über Manage Custom Types erstellt werden. Hier können z. B. Werte für Bedingungen im **Condition**-Editor eingegeben werden.

So erstellen Sie einen Projekt-Parameter

1. Öffnen Sie die **Project Parameter Configuration**.
2. Im Bereich **Parameters**, klicken Sie auf **Add**.



3. Geben Sie die Eigenschaften für den Parameter ein. Datenfelder mit * sind Pflichtfelder:

Datenfelder	Beschreibung
*Name**	Name des Parameters **Hinweis** Um Namenskonflikte zu vermeiden, muss der Name des Parameters eindeutig sein. **Hinweis** Der Name des Projekt-Parameters darf keine Sonderzeichen und Punkte enthalten. Folgende Zeichen sind erlaubt: `^[a-zA-Z][a-zA-Z0-9\-]*$`

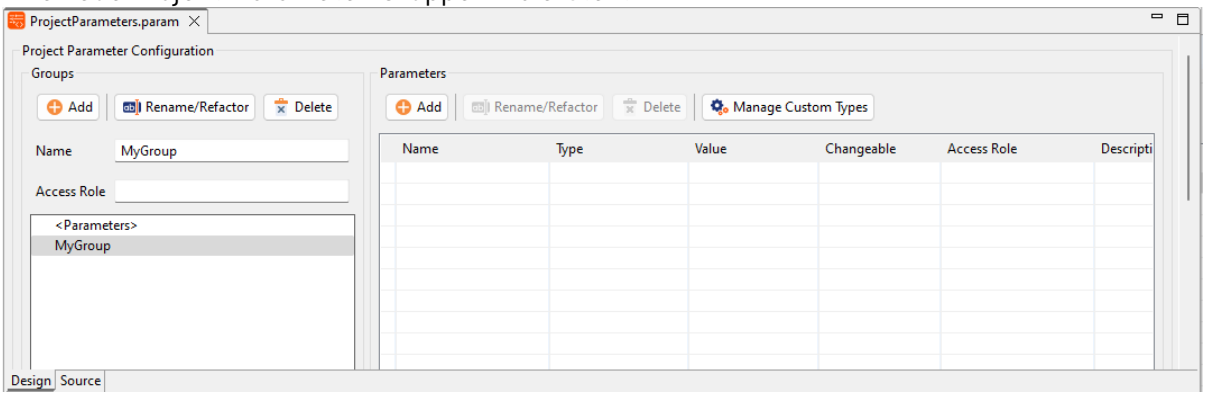
Datenfelder	Beschreibung
*Type**	Datentyp des Parameters **Hinweis** Der Typ des Parameters wird bei Erstellung und beim Überschreiben zur Laufzeit geprüft. Auf Basis des Parametertyps werden die Parameterwerte in den Projektdateien genutzt. Mögliche Werte: • String • Boolean • Integer • Decimal • Date • DateTime • Password **Hinweis** Parameter vom Typ Passwort werden in der **Design**-Ansicht mit Asterisk und in der **Source**-Ansicht (XML-Struktur) verschlüsselt angezeigt. • **Custom Types** mit **Custom Type Values** (Enums)
*Value**	Defaultwert des Parameters
*Changeable**	Typ des Parameters: **Constant** oder **Variable** **Hinweis** Parameter vom Typ **Constant** können nicht zur Laufzeit überschrieben werden. Projekt-Parameter vom Typ **Variable** können zur Laufzeit überschrieben werden. Somit muss der Wert bei Projekt-Parametern vom Typ Variable nicht befüllt werden. Mögliche Werte (Boolean): • True (Variable) • False (Constant)
Access Role	Rolle mit Zugriffsberechtigung zum Editieren des Parameters
Description	Beschreibung des Parameters

5.6.3.3.3 Projekt-Parameter-Gruppen hinzufügen

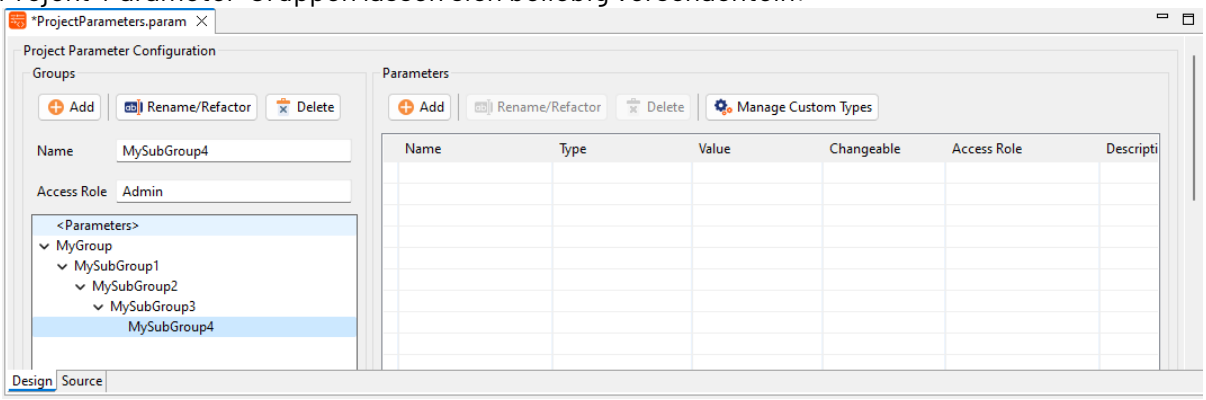
> **Hinweis**
> Um Namenskonflikte zu vermeiden, muss der Name des Parameters eindeutig sein.

So erstellen Sie eine Projekt-Parameter-Gruppe

1. Öffnen Sie die **Project Parameter Configuration**.
2. Im Bereich `Groups`. klicken Sie auf **Add**.
   Eine neue Projekt-Parameter-Gruppe wird erstellt:

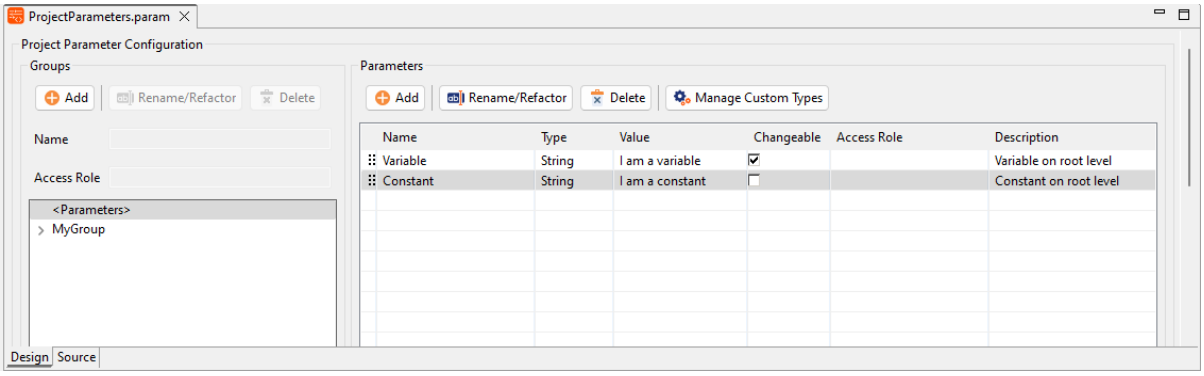


Projekt-Parameter-Gruppen lassen sich beliebig verschachteln:

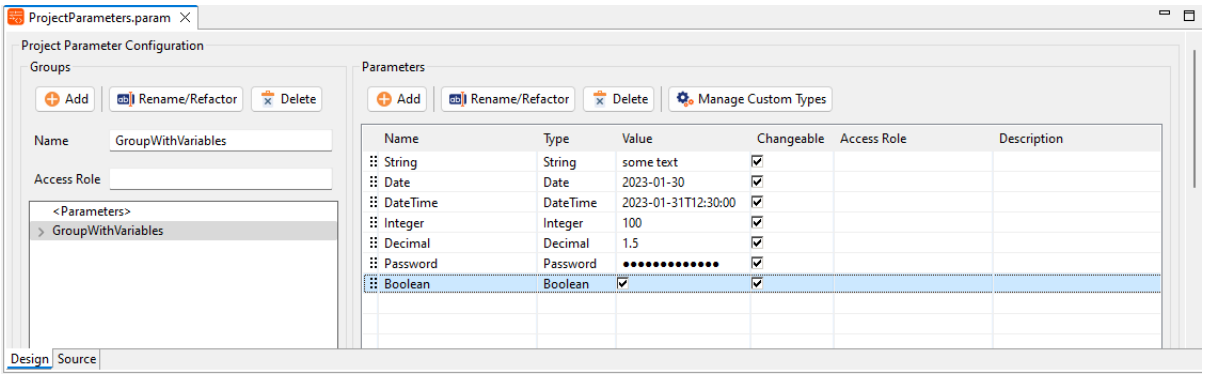


5.6.3.3.4 Beispiel einer Project Parameter Configuration

5.6.3.3.4.1 Project Parameter Configuration mit Constant und Variable auf Root Level



5.6.3.3.4.2 Project Parameter Configuration mit Projekt-Parameter-Gruppe



5.6.3.3.4.3 Beispiel der XML-Struktur

Beispiel für die XML-Struktur einer **Project Parameter Configuration**:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Parameters xmlns="http://softproject.de/parameter/1.0">
    <Enums/>
    <Group name="GroupWithVariables">
        <VariableParameter name="Boolean" type="Boolean">
            <Value>true</Value>
        </VariableParameter>
        <VariableParameter name="Date" type="Date">
            <Value>2023-01-30</Value>
        </VariableParameter>
        <VariableParameter name="DateTime" type="DateTime">
            <Value>2023-01-31T12:30:00</Value>
        </VariableParameter>
        <VariableParameter name="Integer" type="Integer">
            <Value>100</Value>
        </VariableParameter>
        <VariableParameter name="Decimal" type="Decimal">
            <Value>1.5</Value>
        </VariableParameter>
        <VariableParameter name="Password" type="Password">
            <Value>R8hr/okHDzHH1qXCST+aSYJH8auXU86Nhjtw1XVek38sLyFyFM/
KCQM0kVybKJ6EergBHrrtBygJRhWetw==</Value>
        </VariableParameter>
        <VariableParameter name="String" type="String">
            <Value>some text</Value>
        </VariableParameter>
    </Group>
    <ConstantParameter name="Constant" type="String">
        <Value>I am a constant</Value>
    </ConstantParameter>
    <VariableParameter name="Variable" type="String">
        <Value>I am a variable</Value>
    </VariableParameter>
</Parameters>
```

5.6.3.4 Projekt-Parameter umbenennen und verschieben

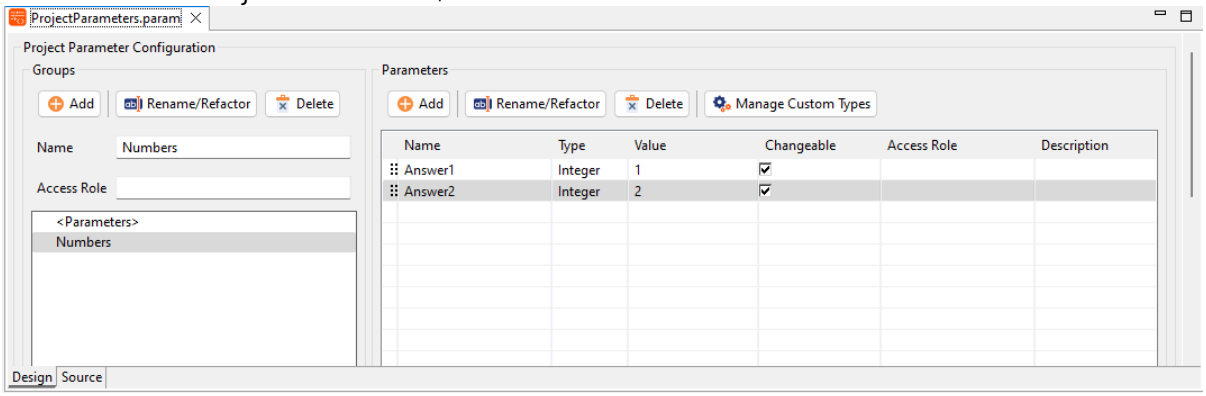
Über die Schaltfläche **Rename/Refactor** können Sie Projekt-Parameter und Gruppen von Projekt-Parametern umbenennen. Diese Umbenennung wird durch Referenzierung (Refactoring) in allen Projekt-Dateien übernommen, in denen die Projekt-Parameter oder Gruppen von Projekt-Parametern verwendet werden.

> **Hinweis**
> Wenn nur der Name des Projekt-Parameters in der Spalte **Name** geändert wird, findet kein Refactoring statt, d. h. der Projekt-Parameter wird nicht in den Dateien im Projekt umbenannt.
> Ein Refactoring kann nur über die Schaltfläche **Rename/Refactor** stattfinden.

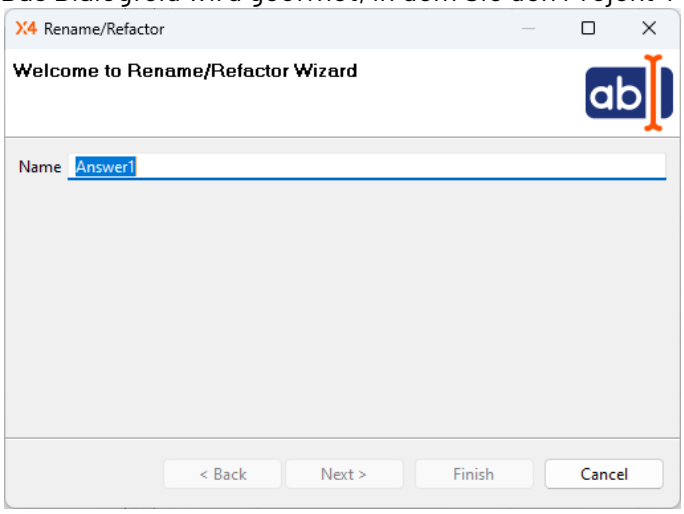
So führen Sie das Refactoring für Projekt-Parameter durch

> **Hinweis**
> Für das Refactoring einer Gruppe führen Sie dieselben Schritte aus.

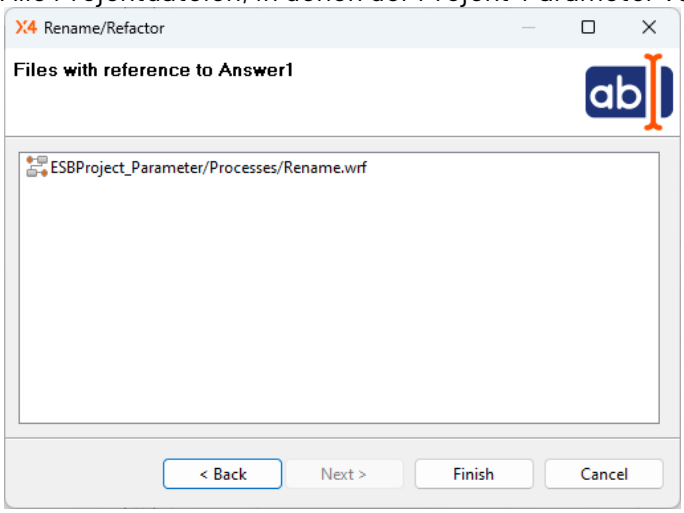
1. Öffnen Sie eine vorhandene `*.param`-Datei (**Project Parameter Configuration**).
2. Wählen Sie den Projekt-Parameter, den Sie umbenennen möchten:



3. Klicken Sie auf **Rename/Refactor**.
   Das Dialogfeld wird geöffnet, in dem Sie den Projekt-Parameter umbenennen können:



4. Geben Sie im Dialogfenster **Rename/Refactor** einen neuen Namen für den Projekt-Parameter ein.

5. Klicken Sie auf **Next**.
   Alle Projektdateien, in denen der Projekt-Parameter verwendet wird, werden angezeigt:


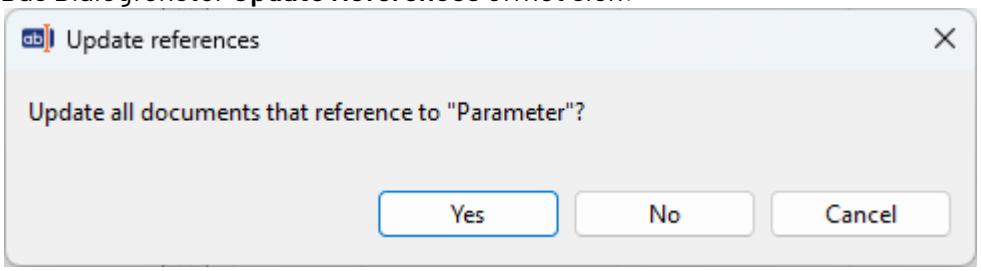
6. Klicken Sie auf **Finish**, um den Projekt-Parameter an allen im Projekt verwendeten Stellen umzubenennen.
   Der Projekt-Parameter wurde in allen Dateien im Projekt umbenannt.

5.6.3.4.1 Projekt-Parameter in eine andere Projekt-Parameter-Gruppe verschieben

In der **Project Parameter Configuration** können Sie einen oder mehrere Projekt-Parameter per Drag&Drop in eine andere Gruppe verschieben und optional neu referenzieren.

Ein Dialogfenster öffnet sich mit der Abfrage, ob ein Refactoring des Projekt-Parameters stattfinden soll oder nicht.

So verschieben Sie Projekt-Parameter in eine andere Gruppe

1. Öffnen Sie eine vorhandene `*.param`-Datei (**Project Parameter Configuration**).
2. Ziehen den Projekt-Parameter, den Sie in eine andere Gruppe verschieben möchten, per Drag&Drop auf die gewünschte Gruppe.
3. Das Dialogfenster **Update References** öffnet sich:


4. Bestätigen Sie mit **Yes**, um ein Refactoring für den gewählten Projekt-Parameter durchzuführen. Beim Refactoring wird der gewählte Projekt-Parameter umbenannt und neu referenziert.
   Der Projekt-Parameter wurde in die Gruppe verschoben und in allen Dateien (Prozesse, Mappings, Conditions, Adapter) im Projekt umbenannt.

5.6.3.5 Custom Types erstellen

In der **Project Parameter Configuration** können Sie **Custom Types** (Enums) mit **Custom Type Values** (Enum-Werten) deklarieren, die Sie später beim Erstellen eines Projekt-Parameters auswählen können.

Custom Types können über **Manage Custom Types** erstellt werden.

Möglicher Anwendungsfall:

- **Custom Type Values** als Auswahloptionen im **Condition Editor**, siehe Bedingungen erstellen und verwenden.

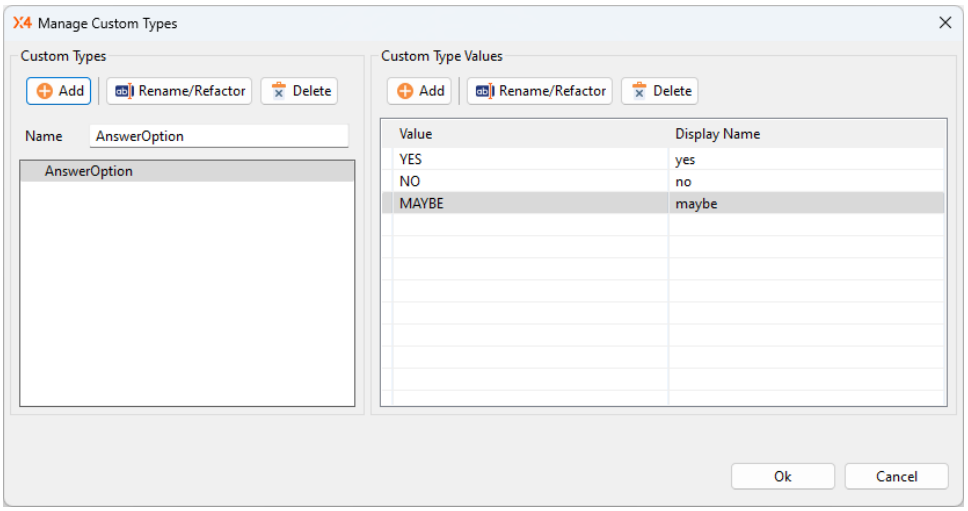
Inhalt dieser Seite:

- Custom Types mit Custom Type Values erstellen
  - Custom Types und Custom Type Values löschen
    - Custom Type löschen
    - Custom Type Value löschen
  - Projekt-Parameter mit Custom Types umbenennen

5.6.3.5.1 Custom Types mit Custom Type Values erstellen

Anwendungsbeispiel:

Wir erstellen in unserem Beispiel den **Custom Type** `AnswerOption` mit den Werten (**Values**) `YES`, `NO`, `MAYBE` und den Anzeigenamen (**Display Names**) `yes`, `no`, `maybe`:

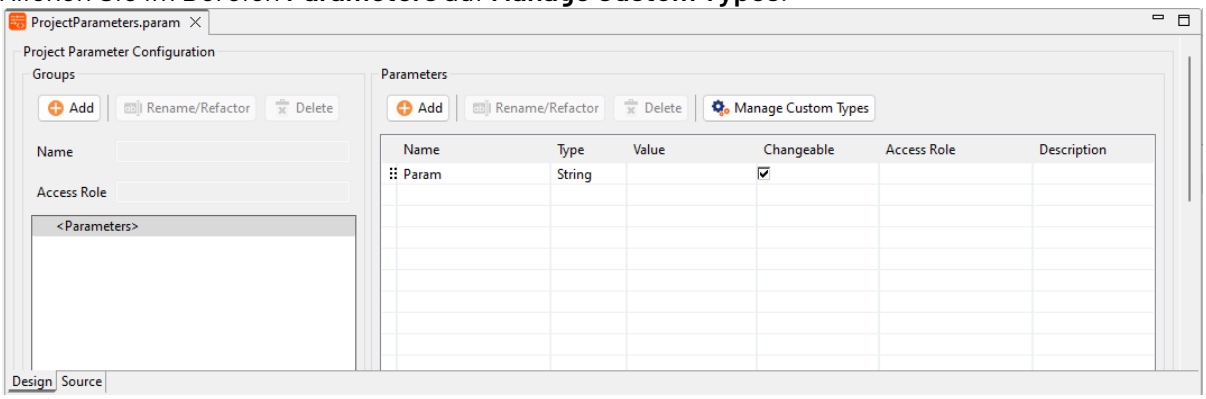


So öffnen Sie das Dialogfenster Manage Custom Types

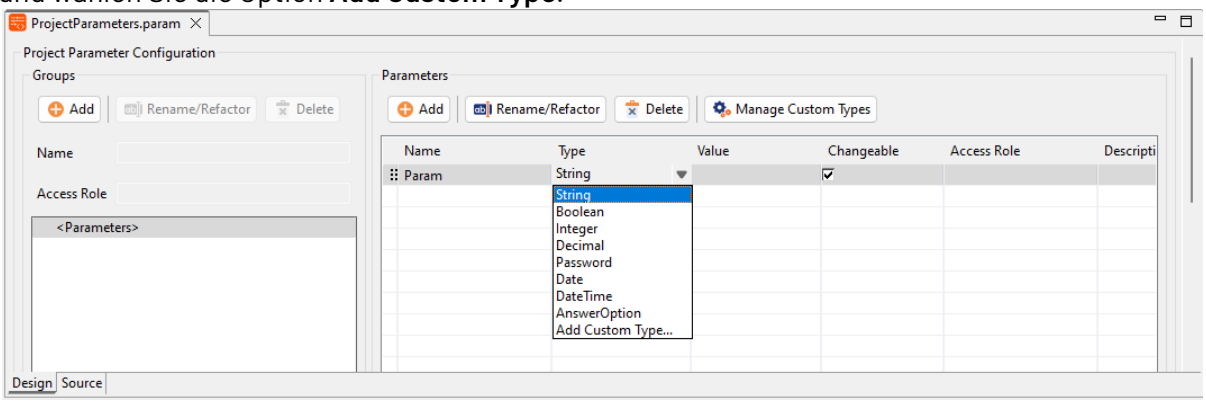
Öffnen Sie eine vorhandene `*.param`-Datei (**Project Parameter Configuration**).

Es gibt zwei Möglichkeiten, wie Sie benutzerdefinierte Projekt-Parameter hinzufügen können:

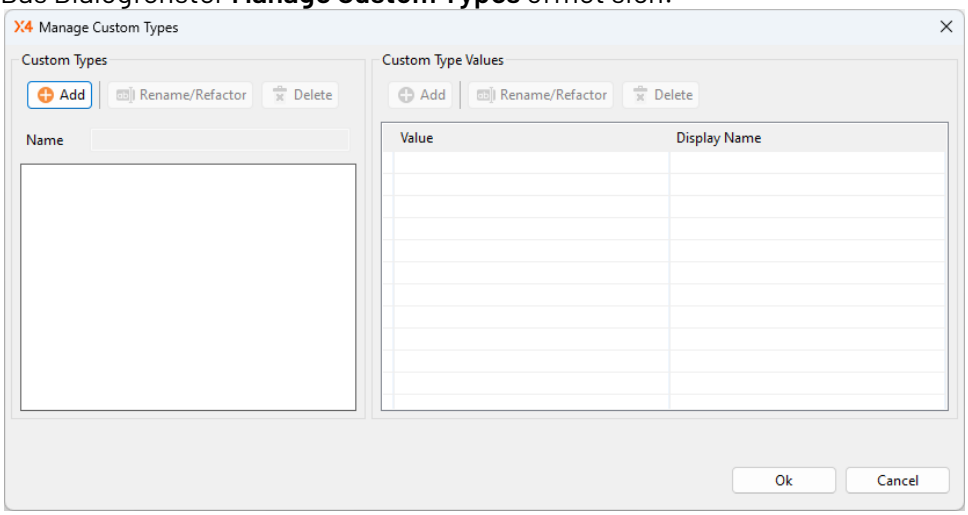
1. Klicken Sie im Bereich **Parameters** auf **Manage Custom Types**.



2. Öffnen Sie in der Spalte **Type** des neu zu erstellenden Projekt-Parameters die Dropdown-Liste und wählen Sie die Option **Add Custom Type**.



Das Dialogfenster **Manage Custom Types** öffnet sich:



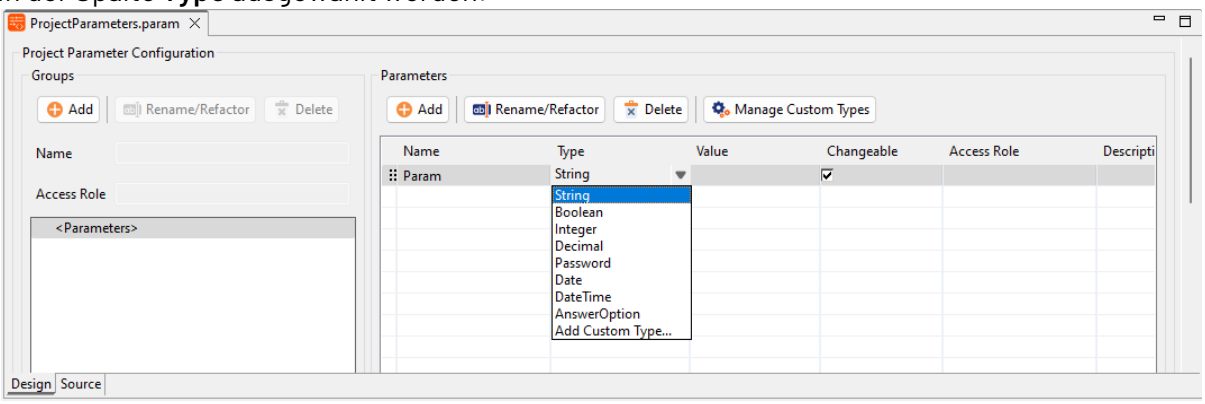
So erstellen Sie Custom Types mit Custom Type Values

1. Klicken Sie im Bereich **Custom Types** auf **Add**, um einen neuen benutzerdefinierten Projekt-Parameter hinzuzufügen.
   Ein neuer benutzerdefinierter Projekt-Parameter-Typ mit dem vordefinierten Namen **CustomType** wurde erstellt.

   > **Hinweis**
   > Der Name des **Custom Type** muss eindeutig sein.

2. Im Bereich **Custom Type Values** können Sie die auswählbaren Werte (**Values**) für die **Custom Types** mit Anzeigenamen (**Display Name**) hinzufügen.
3. Klicken Sie auf **Add**, um einen neuen Wert (**Value**) mit Anzeigename (**Display Name**) hinzuzufügen.
4. Klicken Sie auf **OK**, um die **Custom Types** mit den definierten Werten zu speichern.
   Ein neuer **Custom Type** mit **Custom Type Values** wurde erstellt.

Der neue **Custom Type** mit **Custom Type Values** kann nun bei der Erstellung neuer Projekt-Parameter in der Spalte **Type** ausgewählt werden:



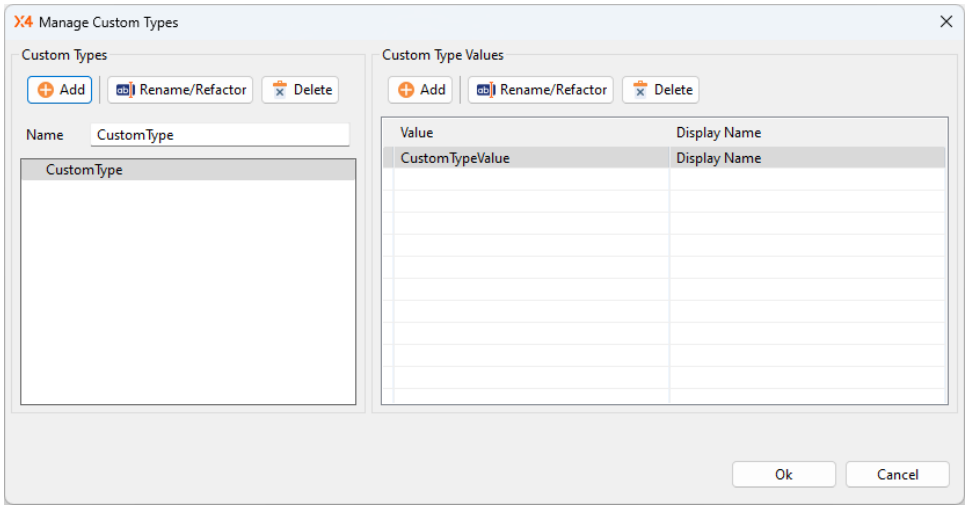
Feld	Beschreibung	Name in Source (XML)
Custom Type Name	Name des **Custom Type**, nach Erstellung des **Custom Type** auswählbar in der Spalte **Type** als neuer Typ mit Auswahloptionen (**Values**)	`Enum name`
Custom Type Value – Value	Auswahloptionen (**Values**), auswählbar in der Spalte **Value** (Werte) bei der Erstellung eines neuen Projekt-Parameters	`<Value>yes <Value>`
Attribut: Custom Type Value – Display Name	Anzeigename (**Display name**) des Werts (**value**)	`displayName` Attribut: Custom Type Value – Display Name

```
<Parameters xmlns="http://softproject.de/parameter/1.0">
    <Enums>
        <Enum name="AnswerOption">
            <Value displayName="yes">YES</Value>
            <Value displayName="no">NO</Value>
            <Value displayName="maybe">MAYBE</Value>
        </Enum>
    </Enums>
    <VariableParameter name="Answer" type="AnswerOption">
        <Value>YES</Value>
    </VariableParameter>
</Parameters>
```

So ändern Sie den Namen im Bereich Custom Types

> **Hinweis**
> Sie können den Namen im Bereich **Custom Types** im Feld **Name** ändern.

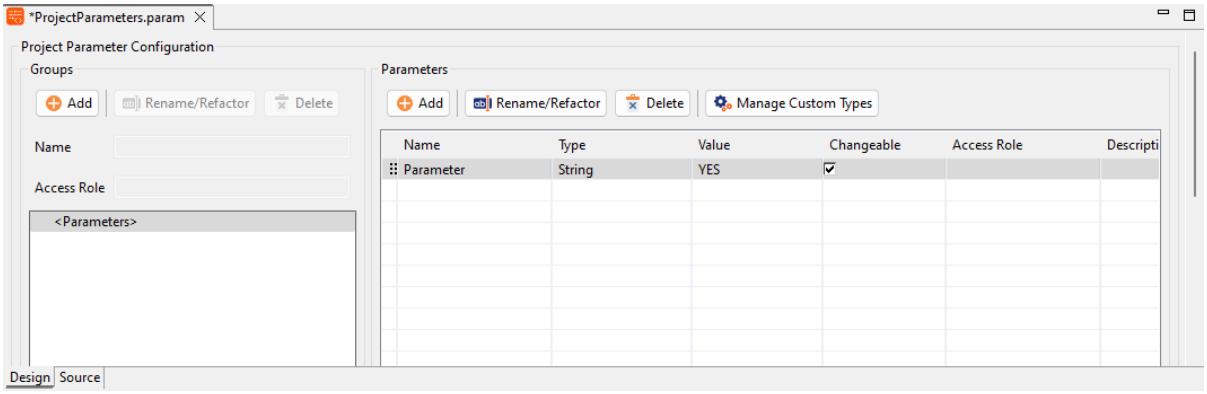
1. Klicken Sie auf den **Custom Type**.
2. Bearbeiten Sie den Namen im Feld **Name**:



5.6.3.5.1.1 Custom Types und Custom Type Values löschen

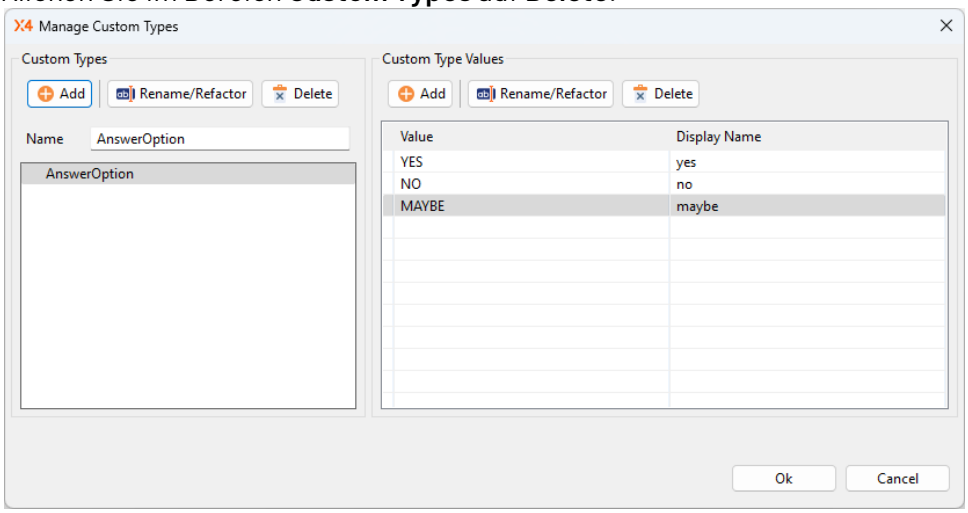
Custom Type löschen
Was passiert, wenn ich einen **Custom Type** lösche, der in Verwendung ist?

Wenn ein benutzerdefinierter Projekt-Parameter-Typ (**Custom Type**) gelöscht wird, wird der **Type** auf String gesetzt, aber der zuvor gesetzte Wert bleibt erhalten:

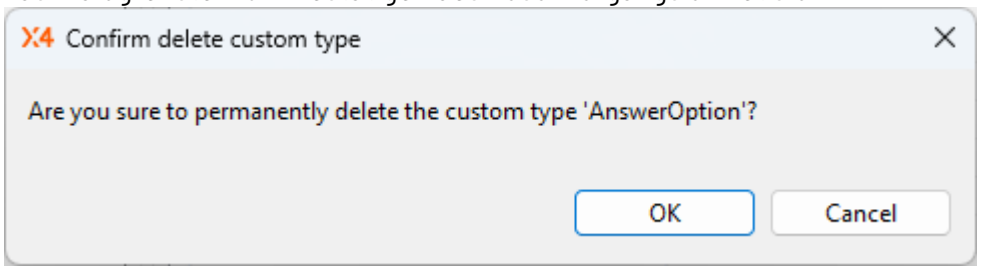


So löschen Sie einen Custom Type:

1. Öffnen Sie eine vorhandene `*.param`-Datei (**Project Parameter Configuration**).
2. Klicken Sie auf die Schaltfläche **Manage Custom Types**.
3. Wählen Sie im Bereich **Custom Types** den **Custom Type**, den Sie löschen möchten.
4. Klicken Sie im Bereich **Custom Types** auf **Delete**.



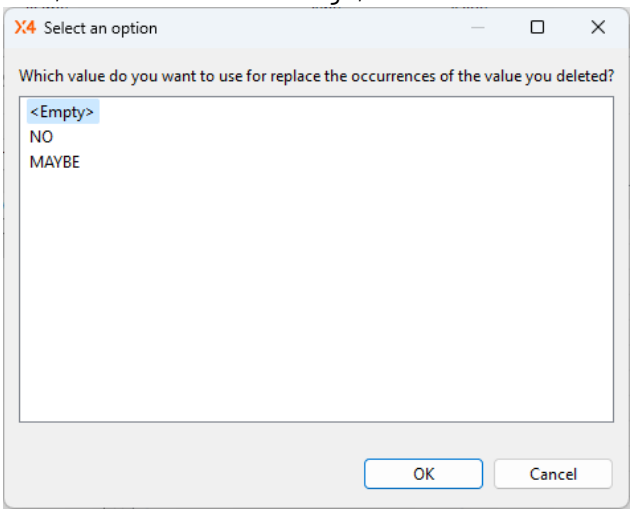
5. Das Dialogfenster zum Bestätigen des Löschvorgangs öffnet sich:



6. Klicken Sie auf **OK, um** das Löschen des **Custom Type** zu bestätigen.
   Der **Custom Type** wurde gelöscht.

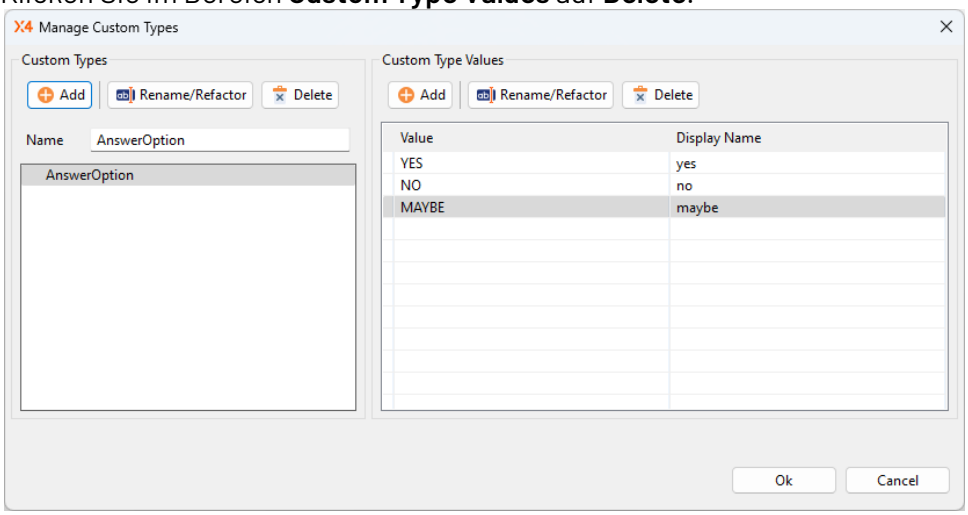
Custom Type Value löschen
Was passiert, wenn ich einen **Custom Type Value** lösche, der in Verwendung ist?

Bevor ein Wert eines benutzerdefinierten Projekt-Parameter-Typs (**Custom Type Value**) gelöscht wird, erscheint eine Abfrage, welcher Wert anstelle gesetzt werden soll.

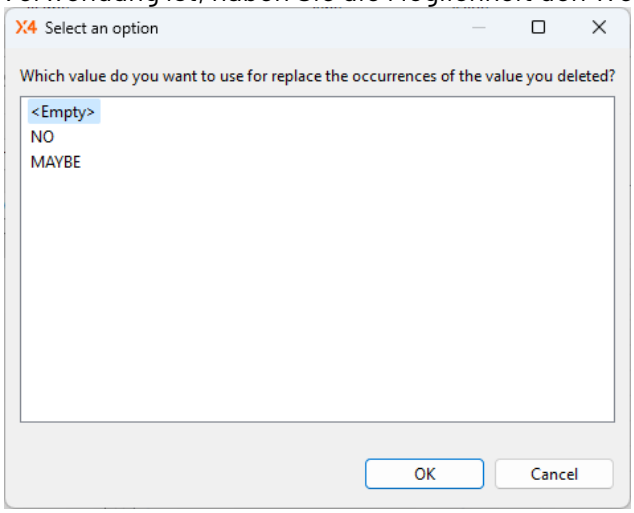


So löschen Sie einen Custom Type Value:

1. Öffnen Sie eine vorhandene `*.param`-Datei (**Project Parameter Configuration**).
2. Klicken Sie auf die Schaltfläche **Manage Custom Types**.
3. Wählen Sie im Bereich **Custom Type Values** den **Custom Type Value**, den Sie löschen möchten.
4. Klicken Sie im Bereich **Custom Type Values** auf **Delete**.



5. Das Dialogfenster zum Bestätigen des Löschvorgangs öffnet sich. Wenn der Wert in Verwendung ist, haben Sie die Möglichkeit den Wert durch einen anderen zu ersetzen:



6. Wählen Sie den Wert, der zum Ersetzen genutzt werden soll.
7. Klicken Sie auf **OK, um** das Löschen des **Custom Type Value** zu bestätigen.
   Der **Custom Type Value** wurde gelöscht und der zum Ersetzen gewählte Wert wurde verwendet.

5.6.3.5.1.2 Projekt-Parameter mit Custom Types umbenennen

Das Umbenennen von Projekt-Parametern mit Custom Types erfolgt wie das Umbenennen von Projekt-Parametern, siehe: Projekt-Parameter umbenennen und verschieben

5.6.3.6 Globale und System-Parameter wiederverwenden

Globale und System-Parameter können in der **Project Parameter Configuration** `*.param` wiederverwendet werden.

Wann ist die Wiederverwendung von globalen und System-Parametern sinnvoll?

Die Wiederverwendung ist sinnvoll, damit Parameter nicht immer wieder neu definiert werden müssen, sondern bestehende Parameter wiederverwendet werden können, z. B. in **Mappings** oder als Parameter bzw. **Component-Parameter** an einem Adapter.

> **Hinweis**
> Um Namenskonflikte zu vermeiden, muss der Name von Parametern und Parameter-Gruppen innerhalb einer Ebene (auf Rootlevel und bei verschachtelten Gruppen) eindeutig sein.

> **Hinweis**
> Folgendes ist bei der Wiederverwendung von Parametern in der **Project Parameter Configuration** `*.param` zu beachten:
>
> - Constants dürfen nur in anderen Constants wiederverwendet werden
> - globale, System- oder Projekt-Parameter dürfen in Variables wiederverwendet werden
> - Parameter dürfen keine Zyklen enthalten (z. B. Variable b verwendet den Wert von Variable a und Variable a verwendet den Wert von Variable b)
> - Nur Parameter vom Typ String können wiederverwendet werden

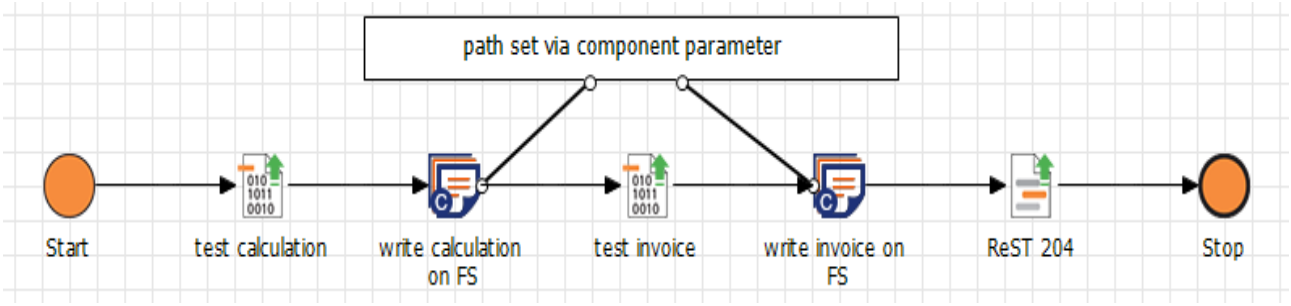
Möglicher Anwendungsfall

Im **File System Adapter** kann ein Pfad über **Component-Parameter** gesetzt werden. Hier kann auf einen Zielordner mit Dateinamen verwiesen werden.

> **Hinweis**
> Anstelle eines **Component-Parameters** kann auch eine andere Art der Parametrierung gewählt werden (z. B. Parameter direkt am Adapter setzen, siehe auch Möglichkeiten der Parametrierung).

Beispiel-Prozess

In diesem Prozess wird jeweils eine Datei in einen Zielordner geschrieben. Der Zielordner und der Dateiname werden über **Component-Parameter** am **File System Adapter** definiert. In den **Component-Parametern** werden globale, System- und Projekt-Parameter wiederverwendet.



Component-Parameter am **File System Adapter** *Write Calculation on FS*:

```
%WORKING_DIR_ABOSLUTE%/%CURRENT_PROJECT%/%filePrefix%_%NOW%.txt
```

Component-Parameter am **File System Adapter** *Write Invoice on FS*:

```
%fileRootPath%/%CURRENT_PROJECT%/%timestampedFile%
```

Im Projekt-Parameter `%fileRootPath%` werden der Projekt-Parameter (Constant) `%X4tempfileDirectory%` und der System-Parameter `%CURRENT_PROJECT%` wiederverwendet.

Im Projekt-Parameter `%timestampedFile%` werden die Projekt-Parameter `%filePrefix%` und `%textfileEnding%` sowie der System-Parameter `%NOW%` wiederverwendet.

Beispiel der XML-Struktur mit wiederverwendeten Parametern in der **Project Parameter Configuration** `*.param`

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Parameters xmlns="http://softproject.de/parameter/1.0">
    <Enums />
    <Group name="FileSystem">
        <ConstantParameter name="X4tempfileDirectory" type="String">
            <Description>This parameter is a constant in a group in this param
file.</Description>
            <Value>%X4tempfileDirectory%\X4</Value>
        </ConstantParameter>
    </Group>
    <ConstantParameter name="tmpDirectory" type="String">
        <Description>This parameter is a constant on root level of this param file.</
Description>
        <Value>C:\Temp</Value>
    </ConstantParameter>
    <ConstantParameter name="filePrefix" type="String">
    <Description>This parameter is a constant on root level of this param file.</
Description>
        <Value>FILE</Value>
    </ConstantParameter>
    <ConstantParameter name="textfileEnding" type="String">
        <Description>This parameter is a constant on root level of this param file.</
Description>
        <Value>.txt</Value>
    </ConstantParameter>
    <VariableParameter name="fileRootPath" type="String">
        <Description>The parameter %fileRootPath% contains the constant parameter
%X4tempfileDirectory% and the system parameter %CURRENT_PROJECT%.</Description>
        <Value>%FileSystem.X4tempfileDirectory%/%CURRENT_PROJECT%</Value>
    </VariableParameter>
    <VariableParameter name="timestampedFile" type="String">
        <Description>The parameter %timestampedFile% contains the system parameter
%NOW% the variable parameters %filePrefix% and %textfileEnding%.</Description>
        <Value>%filePrefix%_%NOW%%textfileEnding%</Value>
    </VariableParameter>
</Parameters>
```

5.7 XSL-Hilfsfunktionen

Welche Hilfsfunktionen von der X4 BPMS zur Verwendung innerhalb von XSL-Mappings bereitstellt werden, erfahren Sie in diesem Abschnitt.

5.7.1 TimeHelper (Veraltet)

⚠ Die Datums-/Uhrzeitfunktionen der Java-Klasse `de.softproject.xsl.TimeHelperDate` zur Verwendung innerhalb von XSL-Mappings sind veraltet und werden in künftigen Versionen der X4 BPMS abgekündigt.

Die Java-Klasse `de.softproject.xsl.TimeHelperDate` befindet sich innerhalb der *X4 Server*-Bibliothek `x4-client.jar` (`<X4>\Server\<wildfly>\modules\system\layers\base\de\softproject\x4\client\main`).

ⓘ Ein XSL-Mapping, das diese Hilfsklasse verwendet, kann nicht in der Transformations-Vorschau des Mapping Editor ausgeführt werden.

5.7.1.1 Datum/Uhrzeit-Methoden

addDay(*java.util.Date* date, *int* amount**)**	Fügt einem `java.util.Date`-Objekt eine beliebige ganzzahlige Anzahl an Tagen hinzu und gibt ein Objekt vom Typ `static java.util.Date` zurück
addHour(*java.util.Date* date, *int* amount**)**	Fügt einem `java.util.Date`-Objekt eine beliebige ganzzahlige Anzahl an Stunden hinzu und gibt ein Objekt vom Typ `static java.util.Date` zurück
addMillisecond(*java.util.Date* date, *int* amount**)**	Fügt einem `java.util.Date` -Objekt eine beliebige ganzzahlige Anzahl an Millisekunden hinzu und gibt ein Objekt vom Typ `static java.util.Date` zurück
addMinute(*java.util.Date* date, *int* amount**)**	Fügt einem `java.util.Date`-Objekt eine beliebige ganzzahlige Anzahl an Minuten hinzu und gibt ein Objekt vom Typ `static java.util.Date` zurück
addMonth(java.util.Date date, int amount)	Fügt einem `java.util.Date`-Objekt eine beliebige ganzzahlige Anzahl an Monaten hinzu und gibt ein Objekt vom Typ `static java.util.Date` zurück

addSecond(*java.util.Date* date, *int* amount**)**	Fügt einem `java.util.Date`-Objekt eine beliebige ganzzahlige Anzahl an Sekunden hinzu und gibt ein Objekt vom Typ `static java.util.Date` zurück
addYear(*java.util.Date* date, *int* amount**)**	Fügt einem `java.util.Date`-Objekt eine beliebige ganzzahlige Anzahl an Jahren hinzu und gibt ein Objekt vom Typ `static java.util.Date` zurück
asMilliseconds(*java.util.Date* date**)**	Gibt ein `java.util.Date`-Objekt als Zeit in Millisekunden seit dem 1.1.1970 als Zahl vom Typ `static long` zurück
canParse(*java.lang.String* format, [*java.lang.String* timezone,] *java.lang.String* value**)**	Überprüft das Format einer Datums mit optionalem Parameter `timezone` für die Zeitzone und gibt ein Objekt vom Typ `static boolean` zurück ? Unknown Attachment Hier wird lediglich das Format und nicht die Gültigkeit des Ergebnisses überprüft.
canParseDefaultFormat([*java.lang.String* timezone,] *java.lang.String* value**)**	Überprüft, ob das verwendete Datumsformat mit optionalem Parameter `timezone` für die Zeitzone dem Standardformat entspricht und gibt ein Objekt vom Typ `static boolean` zurück
createDefaultFormat([*java.lang.String* timezone]**)**	Gibt das aktuelle Datum (mit optionalem Parameter `timezone` für eine spezifische Zeitzone) als Standardformat vom Typ `static java.text.SimpleDateFormat` aus
createFormat(*java.lang.String* format, [*java.lang.String* timezone]**)**	Gibt das aktuelle Datum (mit optionalem Parameter `timezone` für eine spezifische Zeitzone) in einem Format vom Typ `static java.text.SimpleDateFormat` aus ? Unknown Attachment Mehr Informationen zu den Datums-/Zeitformaten siehe http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html
differenceInMilliseconds(*java.util.Date* from, *java.util.Date* to**)**	Berechnet den Unterschied zwischen zwei `java.util.Date` - Objekten in Millisekunden und gibt eine Zahl vom Typ `static long` zurück
format(*java.lang.String* format, [*java.lang.String* timezone,] *java.util.Date* date**)**	Formatiert ein `java.util.Date` - Objekt mit optionalem Parameter `timezone` für die Zeitzone in ein Objekt vom Typ `static java.lang. String`

formatDefaultFormat([*java.lang.String* timezone,] *java.util.Date* date**)**	Formatiert ein `java.util.Date`Objekt mit optionalem Parameter `timezone` für die Zeitzone in ein Standardformat vom Typ `static java.lang.String`
fromMilliseconds(*long* milli**)**	Konvertiert eine Zeitangabe in Millisekunden in e in Objekt vom Typ `static java.util.Date`
getDay(*java.util.Date* date, [*java.lang.String* timezone]**)**	Ermittelt den Tag aus einem `java.util.Date` -Objekt mit optionalem Parameter `timezone` für die Zeitzone, und gibt diesen als Objekt vom Typ `static int` zurück
getDayOfWeek(*java.util.Date* date, [*java.lang.String* timezone]**)**	Ermittelt den Wochentag aus einem `java.util.Date` -Objekt mit optionalem Parameter `timezone` für die Zeitzone, und gibt diesen als Objekt vom Typ `static int` zurück ? Unknown Attachment Die Zählung der Wochentage beginnt bei Sonntag mit 1: • Sonntag: *1* • Montag: *2* • Dienstag: *3* • Mittwoch: 4 • Donnerstag: *5* • Freitag: *6* • Samstag: *7*
getDayOfWeekInMonth(*java.util.Date* date, [*java.lang.String* timezone]**)**	Ermittelt den Wochentag im Monat (z.B. "der wievielte Dienstag im Monat") aus einem `java.util.Date` -Objekt mit optionalem Parameter `timezone` für die Zeitzone, und gibt diesen als g anze Zahl vom Typ `static int` zurück
getDayOfYear(*java.util.Date* date, [*java.lang.String* timezone]**)**	Ermittelt den Tag des Jahres aus einem `java.util.Date` -Objekt mit optionalem Parameter `timezone` für die Zeitzone, und gibt diesen als g anze Zahl vom Typ `static int` zurück
getHour(*java.util.Date* date, [*java.lang.String* timezone]**)**	Ermittelt die Stunden aus einem`java.util.Date`-Objekt mit optionalem Parameter `timezone` die Zeitzone, und gibt diese als ganze Zahl vom Typ `static int` zurück
getMilliSecond(*java.util.Date* date, [*java.lang.String* timezone]**)**	Ermittelt die Millisekunden aus einem `java.util.Date` -Objekt mit optionalem Parameter `timezone` für die Zeitzone, und gibt diese als g anze Zahl vom Typ `static int` zurück

getMinute(*java.util.Date* date, [*java.lang.String* timezone]**)**	Ermittelt die Minuten aus einem `java.util.Date` -Objekt mit optionalem Parameter `timezone` für die Zeitzone , und gibt diese als g anze Zahl vom Typ `static int` zurück
getMonth(*java.util.Date* date, [*java.lang.String* timezone]**)**	Ermittelt den Monat aus einem `java.util.Date` -Objekt mit optionalem Parameter `timezone` für die Zeitzone , und gibt diesen als g anze Zahl vom Typ `static int` zurück ? Unknown Attachment Die Zählung der Monate beginnt mit 0: • Januar: *0* • Februar: *1* • März: *2* • April: *3* • Mai: *4* • Juni: *5* • Juli: *6* • August: *7* • September: *8* • Oktober: *9* • November: *10* • Dezember: *11*
getSecond(*java.util.Date* date, [*java.lang.String* timezone]**)**	Ermittelt die Sekunden aus einem `java.util.Date`-Objekt mit optionalem Parameter `timezone` für die Zeitzone, und gibt diese als g anze Zahl vom Typ `static int` zurück
getWeekOfMonth(*java.util.Date* date, [*java.lang.String* timezone]**)**	Ermittelt die Woche des Monats aus einem `java.util.Date`-Objekt mit optionalem Parameter `timezone` für die Zeitzone , und gibt diese als g anze Zahl vom Typ `static int` zurück
getWeekOfYear(*java.util.Date* date, [*java.lang.String* timezone]**)**	Ermittelt die Kalenderwoche aus einem `java.util.Date`-Objekt mit optionalem Parameter `timezone` für die Zeitzone, und gibt diese als g anze Zahl vom Typ `static int` zurück
getYear(*java.util.Date* date, [*java.lang.String* timezone]**)**	Ermittelt das Jahr aus einem `java.util.Date`-Objekt mit optionalem Parameter `timezone` für die Zeitzone , und gibt dieses als g anze Zahl vom Typ `static int` zurück
isAfter(*java.util.Date* date, *java.util.Date* reference**)**	Überprüft, ob ein `java.util.Date` -Objekt vor einem Referenzobjekt desselben Typs liegt, und gibt ein Objekt vom Typ `boolean` zurück

isBefore(*java.util.Date* date, *java.util.Date* reference**)**	Überprüft, ob sich ein `java.util.Date` -Objekt zeitlich nach einem Referenzobjekt desselben Typs befindet, und gibt ein Objekt vom Typ `boolean` zurück
now()	Ermittelt das aktuelle Datum und gibt dieses als Objekt vom Typ `static java.util.Date` zurück
parse(*java.lang.String format,* [*java.lang.String* timezone,] *java.lang.String* value**)**	Liest eine Zeitangabe in `java.lang.String` -Format mit optionalem Parameter `timezone` für die Zeitzone aus, und gibt ein Objekt vom Typ `static java.util.Date` zurück
parseDefaultFormat([*java.lang.String* timezone,] *java.lang.String* value**)**	Liest eine Zeitangabe in `java.lang.String` -Format mit optionalem Parameter `timezone` für die Zeitzone aus, und gibt ein Objekt vom Typ `static java.util.Date` im Standardformat zurück
setBeginOfDay(*java.util.Date* date, [*java.lang.String* timezone]**)**	Liefert für das eingegebene `java.util.Date` -Objekt (mit optionalem Parameter `timezone` für die Zeitzone) den Beginn des Tages als `static java.util.Date`- Objekt
setBeginOfMonth(*java.util.Date* date, [*java.lang.String* timezone])	Liefert für das eingegebene `java.util.Date`-Objekt (mit optionalem Parameter `timezone` für die Zeitzone) den Beginn des Monats als `static java.util.Date`- Objekt
setBeginOfYear(*java.util.Date* date, [*java.lang.String* timezone]**)**	Liefert für das eingegebene `java.util.Date` -Objekt (mit optionalem Parameter `timezone` für die Zeitzone) den Beginn des Jahres als `static java.util.Date`- Objekt
setDay(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Liefert für das eingegebene `java.util.Date` -Objekt (mit optionalem Parameter `timezone` für die Zeitzone) den Tag des Monats als `static java.util.Date` -Objekt ? Unknown Attachment Ist die eingegeben Zahl höher als die Anzahl der tatsächlichen Tage des Monats, wird die Differenz auf den/die Folgemonat(e) angerechnet.
setDayOfYear(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Liefert für das eingegebene `java.util.Date` -Objekt mit optionalem Parameter `timezone` für die Zeitzone das Datum als `static java.util.Date`- Objekt
setEndOfDay(*java.util.Date* date, [*java.lang.String* timezone]**)**	Liefert für das eingegebene `java.util.Date` -Objekt (mit optionalem Parameter `timezone` für die Zeitzone) das Ende des Tages als `static java.util.Date`- Objekt

setEndOfMonth(*java.util.Date* date, [*java.lang.String* timezone]**)**	Liefert für das eingegebene `java.util.Date` -Objekt (mit optionalem Parameter `timezone` für die Zeitzone) das Ende des Monats als `static java.util.Date`- Objekt
setEndOfYear(*java.util.Date* date, [*java.lang.String* timezone]**)**	Liefert für das eingegebene `java.util.Date` -Objekt (mit optionalem Parameter `timezone` für die Zeitzone) das Ende des Jahres als `static java.util.Date`- Objekt
setHour(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Liefert für das eingegebene `java.util.Date` -Objekt (mit optionalem Parameter `timezone` für die Zeitzone) die Anzahl der Stunden als `static java.util.Date`- Objekt ? Unknown Attachment Ist die eingegebene Zahl größer als 24, wird die Stundendifferenz auf den/die Folgetag(e) angerechnet.
setMillisecond(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Liefert für das eingegebene `java.util.Date` -Objekt (mit optionalem Parameter `timezone` für die Zeitzone) die Anzahl der Millisekunden als `static java.util.Date`-Objekt
setMinute(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Liefert für das eingegebene `java.util.Date` -Objekt (mit optionalem Parameter `timezone` für die Zeitzone) die Anzahl der Minuten als `static java.util.Date`-Objekt ? Unknown Attachment Ist die eingegebene Zahl größer ist als 60, wird die Differenz auf die Folgestunde(n) angerechnet.

setMonth(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Liefert für das eingegebene `java.util.Date` -Objekt mit optionalem Parameter `timezone` für die Zeitzone den Monat als `static java.util.Date`- Objekt ? Unknown Attachment Die Zählung der Monate beginnt mit 0: • Januar: *0* • Februar: *1* • März: *2* • April: *3* • Mai: *4* • Juni: *5* • Juli: *6* • August: *7* • September: *8* • Oktober: *9* • November: *10* • Dezember: *11* Ist die eingegebene Zahl größer als 11, wird die Differenz auf das/die Folgejahr(e) angerechnet.
setSecond(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Liefert für das eingegebene `java.util.Date` -Objekt mit optionalem Parameter `timezone` für die Zeitzone die Sekunden als `static java.util.Date`- Objekt ? Unknown Attachment Ist die eingegebene Zahl größer ist als 60, wird die Differenz auf die Folgeminute(n) angerechnet.
setYear(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Liefert für das eingegebene `java.util.Date` -Objekt mit optionalem Parameter `timezone` für die Zeitzone das Jahr als `static java.util.Date`- Objekt

5.7.1.2 Beispiel

In diesem Beispiel werden stellvertretend für alle Methoden Variablen mit je einem Methodenaufruf initialisiert und deren Rückgabewert in einem entsprechend benannten XML-Element ausgegeben. Hierfür werden im Kopf des XSL-Stylesheets zunächst die Namensräume `x4time` für die TimeHelper-Klasse sowie `sdf` fürs Java-SimpleDateFormat definiert. Zudem werden in Variable `date` und `from_date` zwei Variablen mit einem `java.util.Date`-Objekt für den weiteren Gebrauch als Beispieldaten initialisiert.

```
<xsl:stylesheet xmlns:sdf="java.text.SimpleDateFormat"
xmlns:x4time="de.softproject.xsl.TimeHelperDate" xmlns:xsl="http://www.w3.org/1999/
XSL/Transform" version="1.0">
 <xsl:output indent="yes" method="xml" media-type="text/xml"></xsl:output>
 <xsl:template match="/">

 <!-- Get date/time values -->
    <xsl:variable select="x4time:now()" name="date"></xsl:variable>
    <xsl:variable select="x4time:addDay($date, 1)" name="from_date"></xsl:variable>


 <Output>
        <!-- Output method return values -->
        <now><xsl:value-of select="$date"></xsl:value-of></now>


        <!-- method addDay -->
        <xsl:variable select="x4time:addDay($date, 8)" name="addDayDate"></
xsl:variable>
    <addDay type="date">
             <xsl:value-of select="$addDayDate"></xsl:value-of>
    </addDay>

    <!-- method asMilliseconds -->
    <xsl:variable select="x4time:asMilliseconds($date)" name="asMillisecondsDate"></
xsl:variable>
    <asMilliseconds type="date">
             <xsl:value-of select="$asMillisecondsDate"></xsl:value-of>
    </asMilliseconds>

    <!-- method canParse with Timezone-->
    <xsl:variable select="x4time:canParse('yyyy-MM-dd', 'CET','2013.07.111')"
name="canParseWithTimezone"></xsl:variable>
    <canParse type="withTimezone">
             <xsl:value-of select="$canParseWithTimezone"></xsl:value-of>
    </canParse>

    <!-- method canParseDefaultFormat -->
    <xsl:variable select="x4time:canParseDefaultFormat('2013-07-111 16:56:45.123')"
 name="canParseDefaultFormatSimple"></xsl:variable>
    <canParseDefaultFormat type="simple">
             <xsl:value-of select="$canParseDefaultFormatSimple"></xsl:value-of>
    </canParseDefaultFormat>

    <!-- method createDefaultFormat with timezone -->
    <xsl:variable select="x4time:createDefaultFormat('PST')"
name="sdf_default_withTimezone"></xsl:variable>
    <createDefaultFormatTz type="withTimezone">
             <xsl:value-of select="sdf:format($sdf_default_withTimezone, $date)"></
xsl:value-of>
    </createDefaultFormatTz>

    <!-- method createFormat with timezone -->
```

```
    <xsl:variable select="x4time:createFormat('yyyy-MMM-dd EE','CET')"
name="sdf_simpleWithTimezone"></xsl:variable>
    <createFormatTz type="withTimezone">
             <xsl:value-of select="sdf:format($sdf_simpleWithTimezone, $date)"></
xsl:value-of>
    </createFormatTz>

    <!-- method differenceInMilliseconds: date to date -->
    <xsl:variable select="x4time:differenceInMilliseconds($date, $from_date)"
name="differenceInMillisecondsDate"></xsl:variable>
    <differenceInMilliseconds type="date">
             <xsl:value-of select="$differenceInMillisecondsDate"></xsl:value-of>
    </differenceInMilliseconds>

    <!-- method format with Timezone -->
    <xsl:variable select="x4time:format('yyyy-MM-dd z','CET',$date)"
name="formatWithTimezone"></xsl:variable>
    <formatTZ type="withTimezone">
             <xsl:value-of select="$formatWithTimezone"></xsl:value-of>
    </formatTZ>

    <!-- method formatDefaultFormat with timezone -->
    <xsl:variable select="x4time:formatDefaultFormat('PST',$date)"
name="formatDefaultFormatWithTimezone"></xsl:variable>
    <formatDefaultFormat type="withTimezone">
             <xsl:value-of select="$formatDefaultFormatWithTimezone"></xsl:value-of>
    </formatDefaultFormat>


    <!-- method fromMilliseconds -->
    <xsl:variable select="x4time:fromMilliseconds(1374483531161)"
name="fromMilliseconds"></xsl:variable>
    <fromMilliseconds type="date">
             <xsl:value-of select="$fromMilliseconds"></xsl:value-of>
    </fromMilliseconds>

    <!-- method getDayOfWeek with timezone -->
    <xsl:variable select="x4time:getDayOfWeek($date, PST)"
name="getDayOfWeekWithTimezone"></xsl:variable>
    <getDayOfWeekTZ type="withTimezone">
             <xsl:value-of select="$getDayOfWeekWithTimezone"></xsl:value-of>
    </getDayOfWeekTZ>

    <!-- method isAfter -->
    <xsl:variable select="x4time:isAfter($date, $from_date)" name="isAfter"></
xsl:variable>
    <isAfter type="date">
             <xsl:value-of select="$isAfter"></xsl:value-of>
    </isAfter>

    <!-- method now -->
        <xsl:variable select="x4time:now()" name="now"></xsl:variable>
    <now type="date">
             <xsl:value-of select="$now"></xsl:value-of>
    </now>
```

```
    <!-- method parse with timezone -->
        <xsl:variable select="x4time:parse('yyyy-MM-dd','PST','2013-07-22')"
name="parseWithTimezone"></xsl:variable>
    <parseTZ type="withTimezone">
             <xsl:value-of select="$parseWithTimezone"></xsl:value-of>
    </parseTZ>

    <!-- method parseDefaultFormat with timezone -->
        <xsl:variable select="x4time:parseDefaultFormat('PST','2013-07-22
05:24:30.376')" name="parseDefaultFormatWithTimezone"></xsl:variable>
    <parseDefaultFormatTZ type="withTimezone">
             <xsl:value-of select="$parseDefaultFormatWithTimezone"></xsl:value-of>
    </parseDefaultFormatTZ>

    <!-- method setBeginOfDay -->
    <xsl:variable select="x4time:setBeginOfDay($date)" name="setBeginOfDay"></
xsl:variable>
    <setBeginOfDay type="date">
             <xsl:value-of select="$setBeginOfDay"></xsl:value-of>
    </setBeginOfDay>

        <!-- method setDay with timezone -->
    <xsl:variable select="x4time:setDay($date, 32, PST)"
name="setDayDateWithTimezone"></xsl:variable>
    <setDayTZ type="WithTimezone">
             <xsl:value-of select="$setDayDateWithTimezone"></xsl:value-of>
    </setDayTZ>

    <!-- method setEndOfDay -->
    <xsl:variable select="x4time:setEndOfDay($date)" name="setEndOfDay"></
xsl:variable>
    <setEndOfDay type="date">
             <xsl:value-of select="$setEndOfDay"></xsl:value-of>
    </setEndOfDay>

    <!-- method setMinute with Timezone -->
    <xsl:variable select="x4time:setMinute($date, 72)" name="setMinuteDate"></
xsl:variable>
    <setMinuteTZ type="date">
             <xsl:value-of select="$setMinuteDate"></xsl:value-of>
    </setMinuteTZ>
 </Output>
 </xsl:template>
</xsl:stylesheet>
```

5.7.2 WorkingTimeCalculator (Veraltet)

⚠ Die Arbeitszeit-Berechnungsfunktionen der Java-Klasse `de.softproject.xsl.WorkingTimeCalculator` zur Verwendung innerhalb von XSL-Mappings sind veraltet und werden in künftigen Versionen der X4 BPMS abgekündigt.
Ab Version 6.0.0 der X4 BPMS steht hierfür der Adapter Working Time Calculator zur Verfügung.

Die Java-Klasse `de.softproject.xsl.WorkingTimeCalculator` befindet sich innerhalb der X4 Server-Bibliothek `x4-client.jar` (`<X4>\Server\<wildfly>\modules\system\layers\base\de\softproject\x4\client\main`).

ⓘ Ein XSL-Mapping, das diese Hilfsklasse verwendet, kann nicht in der Transformations-Vorschau des Mapping Editors ausgeführt werden.

5.7.2.1 Methoden zum Definieren der Arbeitszeit

Bevor die unten genannten Methoden für die Arbeitszeitberechnung verwendet werden können, ist es erforderlich, für das Objekt `WorkingTimeCalculator` die gewünschten Arbeitszeiten (Beginn und Ende des Arbeitstages, Anzahl der Wochenarbeitstage, Urlaubstage etc.) über folgende Methoden zu definieren.

setHolidays(*java.lang.String* holidays**)**	Setzt für ein `WorkingTimeCalculator`-Objekt einen oder mehrere Feiertage als Zeichenkette gemäß `java.lang.String` nach dem Muster `yyyy-mm-dd, yyyy-mm-dd, ...` und gibt ein Objekt vom Typ `WorkingTimeCalculator` zurück *Beispiel:* `setHolidays('2013-12-24, 2013-12-25, 2013-12-26'), setHolidays('2016-4-15, 2016-4-16, 2016-4-18')`

setSpecialDays(*java.lang.String* specialDays**)**	Setzt für ein `WorkingTimeCalculator`-Objekt einen Tageszeitraum als Zeichenkette gemäß `java.lang.String` nach dem Muster `yyyy-mm-dd hh:MM-hh:MM, yyyy-mm-dd hh:MM-hh:MM, ...` und gibt ein Objekt vom Typ `WorkingTimeCalculator` zurück Dieser Tageszeitraum wird in jedem Fall als Arbeitszeit berechnet, besitzt also Vorrang gegenüber Feiertagen und Arbeitszeiten. *Beispiel:* `setSpecialDays('2010-12-06 11:30-12:0,2010-01-06 10:30-15:30')`
setWorkingDayBegin(*int* hr, *int* min**)**	Setzt für ein `WorkingTimeCalculator`-Objekt den Beginn des Arbeitstags mit Stunden und Minuten jeweils als Ganzzahl und gibt ein Objekt vom Typ `WorkingTimeCalculator` zurück *Beispiel:* `setWorkingDayBegin(9,0)`
setWorkingDayBegin(*java.lang.String* hr, *java.lang.String* min**)**	Setzt für ein `WorkingTimeCalculator`-Objekt den Beginn des Arbeitstags mit Stunden und Minuten jeweils als Zeichenkette vom Typ `java.lang.String` und gibt ein Objekt vom Typ `WorkingTimeCalculator` zurück *Beispiel:* `setWorkingDayBegin('09','00')`
setWorkingDayEnd(*int* hr, *int* min**)**	Setzt für ein `WorkingTimeCalculator`-Objekt das Ende des Arbeitstags mit Stunden und Minuten jeweils als Ganzzahl und gibt ein Objekt vom Typ `WorkingTimeCalculator` zurück *Beispiel:* `setWorkingDayEnd(17,0)`
setWorkingDayEnd(*java.lang.String* hr, *java.lang.String* min**)**	Setzt für ein `WorkingTimeCalculator`-Objekt das Ende des Arbeitstags mit Stunden und Minuten jeweils als Zeichenkette vom Typ `java.lang.String` und gibt ein Objekt vom Typ `WorkingTimeCalculator` zurück *Beispiel:* `setWorkingDayEnd('17','00')`

setWorkingDays(*java.lang.String* days**)**	Setzt für ein `WorkingTimeCalculator`-Objekt eine Liste der Arbeitstage pro Woche als Zeichenkette vom Typ `java.lang.String` nach dem Muster `<Tagnummer>,<Tagnummer>,...` und gibt ein Objekt vom Typ `WorkingTimeCalculator` zurück Wochentage werden gemäß `java.util.Calendar` als nummerische Liste angegeben, beginnend bei `1` mit dem Sonntag. *Beispiel:* `setWorkingDays('2,3,4,5,6')`

5.7.2.2 Arbeitszeit-Berechnungsfunktionen

addWorkingTimeMinutes(*WorkingTimeCalculator* wtc, *java.util.Date* from, *int* minutes**)**	Fügt einem `java.util.Date`-Objekt eine beliebige ganzzahlige Anzahl an Minuten innerhalb der definierten Arbeitszeiten hinzu und gibt ein Objekt vom Typ `static java.util.Date` zurück *Voraussetzung:* ein Objekt `WorkingTimeCalculator` wurde bereits korrekt initialisiert und als erstes Argument innerhalb einer XSL-Variable übergeben, siehe Methoden zum Definieren der Arbeitszeit
addWorkingTimeMinutesString(*WorkingTimeCalculator* wtc, *java.util.Date* from, *int* minutes)	Fügt einem `java.util.Date`-Objekt eine beliebige ganzzahlige Anzahl an Minuten innerhalb der definierten Arbeitszeiten hinzu und gibt eine Zeichenkette vom Typ `java.lang.String` zurück *Voraussetzung:* ein Objekt `WorkingTimeCalculator` wurde bereits korrekt initialisiert und als erstes Argument innerhalb einer XSL-Variable übergeben, siehe Methoden zum Definieren der Arbeitszeit ? Unknown Attachment Nur mit *Xalan* 2.7.1 kompatibel, nicht mit *Saxon* 9!

addWorkingTimeMinutesString(*WorkingTimeCalculator* wtc, *java.lang.String* from, *int* minutes**)**	Fügt einer Zeichenkette gemäß `java.lang.String` eine beliebige ganzzahlige Anzahl an Minuten innerhalb der definierten Arbeitszeiten hinzu und gibt eine Zeichenkette vom Typ `java.lang.String` zurück *Voraussetzung:* ein Objekt `WorkingTimeCalculator` wurde bereits korrekt initialisiert und als erstes Argument innerhalb einer XSL-Variable übergeben, siehe Methoden zum Definieren der Arbeitszeit ? Unknown Attachment Nur mit *Xalan* 2.7.1 kompatibel, nicht mit *Saxon* 9! *Beispiel:* `addWorkingTimeMinutesString('Thu Jul 11 14:29:24 CEST 2013', 'Mon Jul 18 16:04:55 CEST 2013',200)`
dateDiff(*java.util.Date* from, *java.util.Date* to**)**	Vergleicht zwei `java.util.Date`-Objekte miteinander und gibt die Differenz in Millisekunden als Zahl vom Typ `static long` zurück ? Unknown Attachment Nur mit *Xalan* 2.7.1 kompatibel, nicht mit *Saxon* 9!
dateDiffWorkingMinutes(*WorkingTimeCalculator* wtc, *java.util.Date* from, *java.util.Date* to**)**	Vergleicht zwei `java.util.Date`-Objekte miteinander und gibt die Differenz innerhalb der definierten Arbeitszeiten in Millisekunden als Zahl vom Typ `long` zurück *Voraussetzung:* ein Objekt `WorkingTimeCalculator` wurde bereits korrekt initialisiert und als erstes Argument innerhalb einer XSL-Variable übergeben, siehe Methoden zum Definieren der Arbeitszeit ? Unknown Attachment Nur mit *Xalan* 2.7.1 kompatibel, nicht mit *Saxon* 9!
getOstern(*double* year**)**	Ermittelt das Osterdatum für ein Jahr, das als vierstellige Zahl vom Typ `double` übergeben wird, und gibt eine Zeichenkette vom Typ `java.lang.String` zurück *Beispiel:* `getOstern(2014)`

getWorkingTimeMinutes(*WorkingTimeCalculator* wtc, *java.util.Date* from, *java.util.Date* to**)**	Ermittelt die Anzahl der Minuten innerhalb der definierten Arbeitszeiten, die zwischen einem Start- und einem Enddatum jeweils vom Typ `java.util.Date` liegen, und gibt eine Zahl vom Typ `double` zurück *Voraussetzung:* ein Objekt `WorkingTimeCalculator` wurde bereits korrekt initialisiert und als erstes Argument innerhalb einer XSL-Variable übergeben, siehe Methoden zum Definieren der Arbeitszeit
getWorkingTimeMinutes(*WorkingTimeCalculator* wtc, *int* fromYear, *int* fromMonth, *int* fromDay, *int* fromHr, *int* fromMin**)**	Ermittelt die Anzahl der Minuten innerhalb der definierten Arbeitszeiten, die zwischen einem Startdatum (definiert als Ganzzahlen für Jahr, Monat, Tag, Stunde und Minute) und dem aktuellen Zeitpunkt liegen, und gibt eine Zahl vom Typ `double` zurück *Voraussetzung:* ein Objekt `WorkingTimeCalculator` wurde bereits korrekt initialisiert und als erstes Argument innerhalb einer XSL-Variable übergeben, siehe Methoden zum Definieren der Arbeitszeit
getWorkingTimeMinutes(*WorkingTimeCalculator* wtc, *int* fromYear, *int* fromMonth, *int* fromDay, *int* fromHr, *int* fromMin, *java.util.Date* to**)**	Ermittelt die Anzahl der Minuten innerhalb der definierten Arbeitszeiten, die zwischen einem Startdatum (definiert als Ganzzahlen für Jahr, Monat, Tag, Stunde und Minute) und einem Enddatum vom Typ `java.util.Date` liegen, und gibt eine Zahl vom Typ `double` zurück *Voraussetzung:* ein Objekt `WorkingTimeCalculator` wurde bereits korrekt initialisiert und als erstes Argument innerhalb einer XSL-Variable übergeben, siehe Methoden zum Definieren der Arbeitszeit
getWorkingTimeMinutes(*WorkingTimeCalculator* wtc, *int* fromYear, *int* fromMonth, *int* fromDay, *int* fromHr, *int* fromMin, *int* toYear, *int* toMonth, *int* toDay, *int* toHr, *int* toMin**)**	Ermittelt die Anzahl der Minuten innerhalb der definierten Arbeitszeiten, die zwischen einem Start- und Enddatum (jeweils definiert als Ganzzahlen für Jahr, Monat, Tag, Stunde und Minute) liegen, und gibt eine Zahl vom Typ `double` zurück *Voraussetzung:* ein Objekt `WorkingTimeCalculator` wurde bereits korrekt initialisiert und als erstes Argument innerhalb einer XSL-Variable übergeben, siehe Methoden zum Definieren der Arbeitszeit

5.7.2.3 Beispiel

```
<xsl:stylesheet xmlns:x4time="de.softproject.xsl.TimeHelperDate"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:x4wtc="de.softproject.xsl.WorkingTimeCalculator"
  exclude-result-prefixes="x4time x4wtc" version="1.0">
  <xsl:output indent="yes" method="xml" media-type="text/xml"></xsl:output>
  <xsl:template match="/">
    <Output>
      <!-- Create new WorkingTimeCalculator object and store it in variable WTC -->
      <xsl:variable select="x4wtc:new()" name="WTC1"></xsl:variable>

      <!-- Get a java.util.date object with current date/time -->
      <xsl:variable select="x4time:now()" name="date"></xsl:variable>
      <now><xsl:value-of select="$date"></xsl:value-of></now>

      <!-- Set 2013 holidays to new year's eve and christmas -->
      <xsl:variable select="x4wtc:setHolidays($WTC1,'2013-1-1, 2013-12-25')"
name="WTC2"></xsl:variable>

      <!-- Set extra working time to December 24th from  (Santa's X-mas delivery
shift) -->
      <xsl:variable select="x4wtc:setSpecialDays($WTC2,'2013-12-24 23:0-23:59')"
 name="WTC3"></xsl:variable>

      <!-- Set working day begin time to 9:00 AM for the WTC object variable -->
      <xsl:variable select="x4wtc:setWorkingDayBegin($WTC3,9,0)" name="WTC4"></
xsl:variable>

      <!-- Set working day end time to 5:00 PM (17:00) for the WTC object variable --
>
      <xsl:variable select="x4wtc:setWorkingDayEnd($WTC4, 17, 0)" name="WTC5"></
xsl:variable>

      <!-- Set working days to Monday, Tuesday, Wednesday, Thursday, Friday for the
WTC object variable -->
      <xsl:variable select="x4wtc:setWorkingDays($WTC5,'2,3,4,5,6')" name="WTC"></
xsl:variable>

      <!-- Add 500 working time minutes to a date -->
      <xsl:variable select="x4wtc:addWorkingTimeMinutes($WTC, $date,500)"
name="addWorkingTimeMinutes"></xsl:variable>
      <addWorkingTimeMinutes><xsl:value-of select="$addWorkingTimeMinutes"></
xsl:value-of></addWorkingTimeMinutes>

      <!-- Get Easter date for year 2014 -->
      <xsl:variable select="x4wtc:getOstern(2014)" name="getOstern"></xsl:variable>
      <getOstern><xsl:value-of select="$getOstern"></xsl:value-of></getOstern>

      <!-- Calculate working minutes between two java.util.Date objects -->
      <xsl:variable select="x4wtc:getWorkingTimeMinutes($WTC,$date,
$addWorkingTimeMinutes)" name="getWorkingTimeMinutes"></xsl:variable>
```

```
      <getWorkingTimeMinutes type="date"><xsl:value-of
select="$getWorkingTimeMinutes"></xsl:value-of></getWorkingTimeMinutes>
    </Output>
  </xsl:template>
</xsl:stylesheet>
```

5.7.3 checkXML

Wie die XML-Prüfungsfunktion `checkXML` aus der Java-Klasse `de.softproject.xsl.XSLHelper` innerhalb von XSL-Mappings verwendet wird

Die Java-Klasse `de.softproject.xsl.XSLHelper` befindet sich innerhalb der X4 Server-Bibliothek `x4-client.jar` (`<X4>\Server\<wildfly>\modules\system\layers\base\de\softproject\x4\client\main`).

> Ein XSL-Mapping, das diese Hilfsklasse verwendet, kann nicht in der Transformations-Vorschau des Mapping Editors ausgeführt werden.

5.7.3.1 Methodenbeschreibung

checkXML(*java.lang.String* URL**)**	Prüft ein Dokument, das über eine beliebige gültige URL als Zeichenkette gemäß `java.lang.String`, referenziert wird, ob es wohlgeformtes XML ist und gibt einen Wert vom Typ `boolean` zurück Gültige URL-Protkolle sind u. a.: • `file:///` für lokale Dateien • `xstore://` für Repository-Dokumente • `context:` für Variablen im Kontext der aktuellen *X4*-Prozessinstanz

5.7.3.2 Beispiel

Im Folgenden XSL-Mapping wird auf ein XML-Dokument, das in einer Prozessvariable *`Var1`* zur Laufzeit eines zugehörigen *X4*-Prozesses hinterlegt wurde, über das `context:`-URL-Protokoll zugegriffen. Mit der eingebundenen Java-Methode `checkXML` erfolgt dabei innerhalb des XSL-Mappings zunächst eine Prüfung auf Wohlgeformtheit, gefolgt von einer Verarbeitung und einer Fehlerbehandlung.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet exclude-result-prefixes="xslhelper" version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xslhelper="de.softproject.xsl.XSLHelper">
  <xsl:output indent="yes" media-type="text/xml" method="xml"/>
  <xsl:template match="/">
    <Output>
      <!-- Checking if Var1 is XML or not  -->
      <!-- If true, output it -->
      <xsl:choose>
        <xsl:when test="xslhelper:checkXML('context:Var1')">
          <!-- Define an XSL variable and assign a process variable "Var1",
                that is accessible as an external document via context: protocol -->
          <xsl:variable name="Var1Content" select="document('context:Var1')"/>
          <xsl:attribute name="isXML">
            <xsl:text>Yes</xsl:text>
          </xsl:attribute>
          <!-- Output XML content of Var1 -->
          <Data>
            <xsl:copy-of select="$Var1Content"/>
          </Data>
        </xsl:when>
        <xsl:otherwise>
          <xsl:attribute name="isXML">
            <xsl:text>No</xsl:text>
          </xsl:attribute>
        </xsl:otherwise>
      </xsl:choose>
    </Output>
  </xsl:template>
</xsl:stylesheet>
```

5.8 XSLT-Funktionen

Im Mapping Editor können Sie die von XSLT 2.0 bereitgestellten Funktionen in Ihren XSL-Mappings einsetzen.
Für weitere Informationen besuchen Sie bitte die Webseite der World Wide Web Consortium (W3C).

5.9 XPath-Funktionen

Welche von XPath 1.0 bereitgestellten Funktionen Sie in XSL-Mappings einsetzen können

> Mit [eckigen Klammern] markierte Argumente sind optional.

5.9.1 Knotengruppen-Funktionen (node-set)

count(Knotengruppe**)**	Gibt die Anzahl der Knoten in `Knotengruppe` als ganze Zahl zurück
id (Id**)**	Wählt den Knoten mit der in `Id` angegebenen eindeutigen ID aus. **Beispiel:** `<xsl:value-of select="id('abc')/child::para"/>` Wählt Kindknoten `para` des Knotens mit der eindeutigen ID *abc* aus.
last ( )	Gibt die Positionsnummer des letzten Knotens im aktuellen Kontext als ganze Zahl zurück. **Beispiel:** `<xsl:if test="position() = last()">` prüft, ob der aktuelle Knoten der letzte Knoten ist.
local-name ([Knotengruppe]**)**	Gibt aus einem Knotennamen mit Namensraumangabe den lokalen Teil des Namens als Zeichenkette zurück. Optional kann als Argument eine `Knotengruppe` ausgewählt werden.
name ([Knotengruppe]**)**	Gibt den Namen des aktuellen Knotens als Zeichenkette zurück. Optional kann als Argument eine `Knotengruppe` ausgewählt werden.
namespace-uri ([Knotengruppe]**)**	Gibt den Namespace-URI (Uniform Resource Identifier) des aktuellen Element- oder Attributknotens als Zeichenkette zurück . Optional kann als Argument eine Knotengruppe ausgewählt werden. **Beispiel:** `<xsl:value-of select="namespace-uri()"/>` Gibt den Namespace-URI (sofern vorhanden) des aktuellen Knotens aus.
position ( )	Gibt die Positionsnummer des aktuellen Knotens als ganze Zahl zurück. Der erste Knoten erhält die Positionsnummer *1*.

5.9.2 Boolesche Funktionen (boolean)

Boolesche XPath-Funktionen geben *true*/*false*-Werte zurück und können mit Vergleichsoperatoren verwendet werden.

boolean (Arg**)**	Prüft ein Argument `Arg` auf seine Wahrheit und gibt einen Booleschen Wert zurück. Alle Argumente, die eine positive Zahl, eine nicht-leere Zeichenkette oder eine nicht-leere Knotengruppe sind, werden als *true* interpertiert.

false ()	Gibt *false* zurück.
lang (Arg**)**	Gibt *true* zurück, wenn die in Argument Arg angegebene Sprache, die in einem xml:lang-Attribut angegeben wurde, mit der Sprachangabe des aktuellen Knotens mindestens zum Teil übereinstimmt. **Beispiel:** lang("en") gibt *true* zurück, wenn der aktuelle Knoten das Attribut xml:lang="en" besitzt oder wie in xml:lang="en-US" teilweise übereinstimmt.
not (Arg**)**	Gibt die Negation eines Booleschen Ausdrucks in Arg zurück. **Beispiel:** <xsl:if test="not(element > 1)"> gibt *true* zurück, wenn der Elementinhalt von element kleiner oder gleich *1* ist.
true ()	Gibt *true* zurück.

5.9.3 Number-Funktionen

XPath-Number-Funktionen geben ganze Zahlen zurück und können mit Vergleichsoperatoren verwendet werden.

ceiling (Arg**)**	Rundet das Argument Arg auf die nächsthöhere ganze Zahl auf und gibt diese zurück (z. B. *0.001* auf *1*). Falls nötig wird zunächst das Argument in eine Zahl konvertiert. Mathematisch korrekt ausgedrückt: Gibt die kleinste ganze Zahl zurück, die nicht kleiner ist als das Argument Arg
floor (Arg**)**	Rundet das Argument Arg auf die nächstniedrigere ganze Zahl ab und gibt diese zurück (z. B. *0.999* auf *0*). Falls nötig wird zunächst das Argument in eine Zahl konvertiert. Mathematisch korrekt ausgedrückt: Gibt die größte ganze Zahl zurück, die nicht größer ist als das Argument Arg
number ([Arg]**)**	Konvertiert den aktuellen Knoteninhalt in eine Zahl und gibt diese zurück. Optional kann ein Argument Arg eine Knotengruppe, eine Zeichenkette, einen Booleschen Wert oder ein anderes Objekt angeben, das konvertiert werden soll. Boolesche Werte *true* und *false* werden zu *1* und *0* konvertiert. **Beispiel:** Eine Zeichenkette *"3xy"* wird in *3* umgewandelt; *"xy"* als *NaN* (Not a Number) interpretiert.

round (Arg**)**	Rundet das Argument `Arg` zu einer ganzen positiven oder negativen Zahl auf oder ab und gibt diese zurück. Zahlen kleiner als *`[Wert].5`* werden abgerundet, bei Bruchgrößen größer oder gleich *`[Wert].5`* wird auf die nächsthöhere ganze Zahl aufgerundet. **Beispiel:** *0.49* wird zu *0* abgerundet, *-1.5* wird zu *-1* aufgerundet.
sum (Knotengruppe**)**	Bildet eine Gesamtsumme aller Knoteninhalte des Arguments `Knotengruppe` und gibt diese wenn möglich als Zahl zurück. Falls nötig werden zunächst die Knoteninhalte in Zahlen konvertiert. Können Werte nicht als Zahl interpretiert werden, dann werden diese als *`NaN`* (Not a Number) interpretiert.

5.9.4 Zeichenketten-Funktionen (String)

XPath-String-Funktionen verarbeiten Objekte und geben Zeichenketten oder bei Zeichenketten-Prüfungen Boolesche Werte zurück.

concat (Arg1, Arg2 [, ...]**)**	Hängt mehrere Zeichenketten, die als Argumente `Arg1`, `Arg2` etc. übergeben werden, aneinander und gibt eine Zeichenkette zurück. Falls nötig werden zunächst die Argumente in Zeichenketten konvertiert.
contains (Arg, Substring**)**	Prüft, ob das Argument `Substring` in Argument `Arg` enthalten ist und gibt einen Booleschen Wert zurück. Falls nötig werden zunächst die Argumente in Zeichenketten konvertiert. **Beispiel:** `<xsl:if test="contains('SoftProject', 'Project')">` gibt den Booleschen Wert *`true`* zurück.
starts-with (Arg, Substring**)**	Prüft, ob das Argument `Arg` mit dem Argument `Substring` beginnt und gibt einen Booleschen Wert zurück. Falls nötig werden zunächst die Argumente in Zeichenketten konvertiert. **Beispiel:** `<xsl:if test="starts-with('SoftProject', 'Soft')">` gibt den Booleschen Wert *`true`* zurück.
normalize-space ([Arg]**)**	Entfernt vom aktuellen Knoteninhalt die Leerzeichen am Anfang und Ende, ersetzt mehrfach aufeinanderfolgende Leerzeichen sowie Tabulatoren und Zeilenumbrüche durch ein einziges Leerzeichen und gibt den Knoteninhalt als Zeichenkette zurück Optional kann ein Argument `Arg` verwendet werden, das falls nötig zunächst in eine Zeichenkette konvertiert wird.

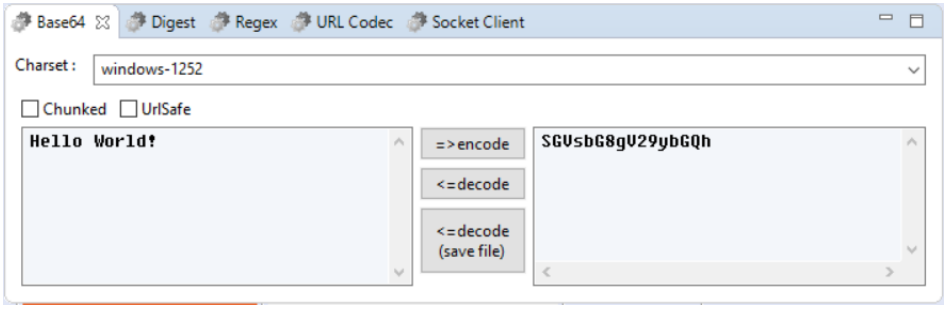
string ([Arg]**)**	Konvertiert den aktuellen Knoteninhalt in eine Zeichenkette und gibt diese zurück. Optional kann ein Argument `Arg` verwendet werden. Boolesche Werte werden als Zeichenkette *`true`* oder *`false`* ausgegeben, Kommazahlen werden nach Möglichkeit vereinfacht. Für erweiterte Konvertierungen von Zahlen zu Zeichenketten die XSLT-Funktion `format-numbers()` verwenden; siehe XSLT-Funktionen. **Beispiel:** `<xsl:value-of select="string(4.00)"/>` gibt die Zeichenkette *`4`* zurück. Prüfen, ob ein Wert eine Zahl ist: `<xsl:if test="string(number(@value))!='NaN'">`.
string-length ([Arg]**)**	Gibt die Anzahl der Zeichen des aktuellen Knoteninhalts als ganze Zahl zurück. Falls nötig wird zunächst das Argument in eine Zeichenkette konvertiert. Optional kann ein Argument `Arg` verwendet werden, das falls nötig zunächst in eine Zeichenkette konvertiert wird.
substring (Arg, Start [, Länge]**)**	Extrahiert aus der Zeichenkette in Argument `Arg` eine Teilzeichenkette, die ab Zeichen `Start` beginnt und eine Zeichenkette zurückgibt. Die Zählung der Zeichen beginnt bei *`1`*. Optional kann das Argument `Länge` verwendet werden, das die maximale Länge der Teilzeichenkette angibt.
substring-before (Arg, Substring**)**	Prüft, ob die Zeichenkette in `Arg` vor der Teilzeichenkette in Argument `Substring` ein oder mehrere Zeichen besitzt und gibt diese als Zeichenkette zurück. Wenn `Substring` nicht in `Arg` enthalten ist, wird eine leere Zeichenkette zurückgegeben. **Beispiel:** `<xsl:value-of select="substring-before('2008/04/01', '/')"/>` gibt *`2008`* zurück.
substring-after (Arg, Substring**)**	Prüft, ob die Zeichenkette in `Arg` hinter der Teilzeichenkette in Argument `Substring` ein oder mehrere Zeichen besitzt und gibt die geänderte Zeichenkette zurück. Wenn `Substring` nicht in `Arg` enthalten ist, wird eine leere Zeichenkette zurückgegeben. **Beispiel:** `<xsl:value-of select="substring-after('2008/04/01', '20')"/>` gibt *`08/04/01`* zurück.
translate (Arg, Substring 1, Substring 2**)**	Sucht in Argument `Arg` nach der Zeichenkette in Argument `Substring1`, ersetzt diese durch die Zeichenkette in Argument `Substring2` und gibt die geänderte Zeichenkette zurück. Falls nötig wird zunächst das Argument `Arg` in eine Zeichenkette konvertiert. Jeder Treffer von `Substring1` wird durch `Substring2` ersetzt. **Beispiel:** `<xsl:value-of select="translate('hallo welt', 'abcdefghijklmnopqrstuvwxyz', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')" />` gibt *`HALLO WELT`* zurück.

5.10 Entwickler-Hilfswerkzeuge

Eine Reihe von Entwickler-Hilfswerkzeugen unterstützen Prozess-Entwickler u.a. bei der schnellen URL-Kodierung, Base64-Konvertierung und der Erzeugung von Hash-Codes. In jeweils einer Sicht innerhalb des X4 Designers lassen sich damit die Funktionen des Eclipse-Plug-ins DevUtils grafisch nutzen.

Um das gewünschte Hilfswerkzeug kann über das Menü **View** aufgerufen werden.

Sicht	Erläuterung
Base64	Hier können Sie eine aus der Zwischenablage eingefügte Zeichenkette Base64-kodieren oder dekodieren



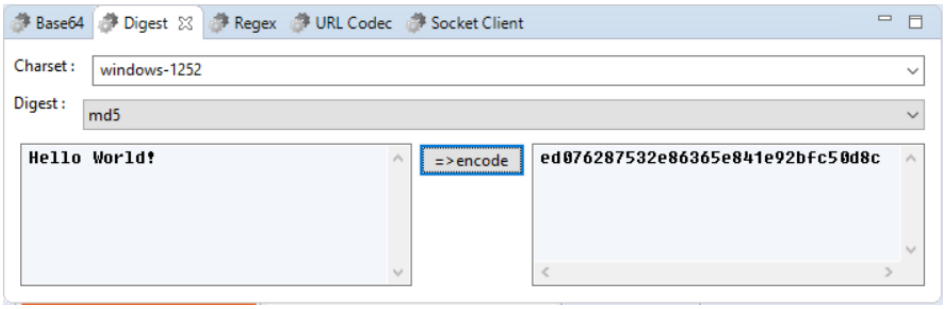
Verfügbare Optionen:

- **Charset:** Zeichensatz für die Zeichenkette
- **Chunked:** Base64-kodierte Zeichenkette blockweise ausgeben
- **UrlSafe:** Beim Base64-Kodieren nur Zeichen verwenden, die in URLs erlaubt sind

Beachten Sie:

- Die Schaltfläche **decode (save file)** speichert die dekodierte Base64-Zeichenkette als Datei in Ihrem lokalen Dateisystem.
- Derzeit lassen sich im X4 Designer nur Dateien aus Ihrem lokalen Dateisystem hineinziehen, jedoch keine Dateien, die per Drag & Drop aus dem **Repository Navigator** eingefügt wurden.

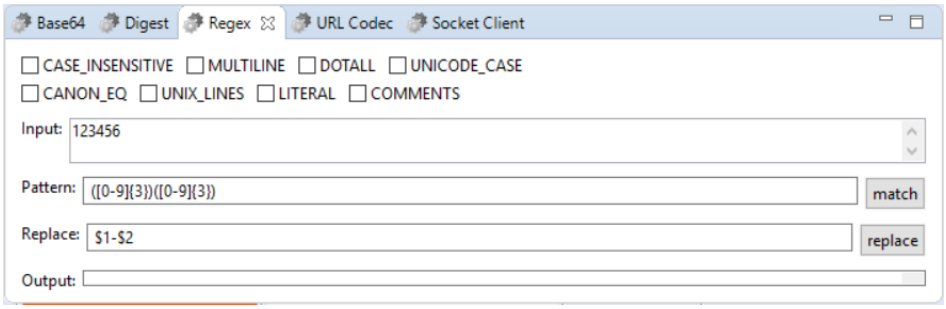
Digest Hier können Sie einen Hash-Wert für eine aus der Zwischenablage eingefügte Zeichenkette erzeugen



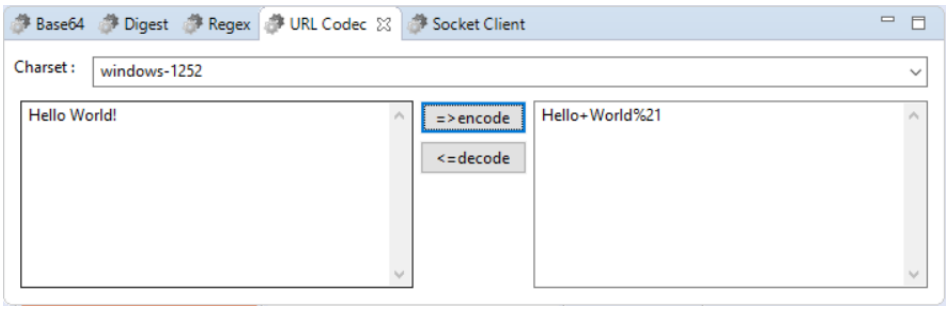
Verfügbare Optionen:

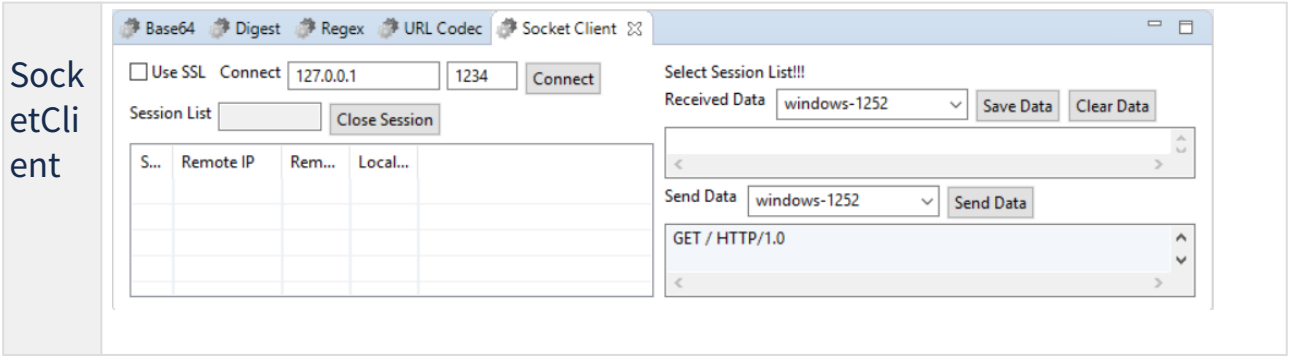
- **Charset:** Zeichensatz für die Zeichenkette
- **Digest:** Algorithmus wählen:
  - `md5`: MD5-Hash erzeugen (Standard)
  - `sha`: Mit dem Secure Hash Algorithm (SHA) einen Hash erzeugen
  - `sha256`: SHA-Hash mit 256 Bit erzeugen
  - `sha384`: SHA-Hash mit 384 Bit erzeugen
  - `sha512`: SHA-Hash mit 512 Bit erzeugen

Regex Hier können Sie einen regulären Ausdruck (*Regex*) auf eine aus der Zwischenablage eingefügte Zeichenkette anwenden, um den Ausdruck zu testen (Schaltfläche **match**) oder um die Zeichenkette gemäß dem Muster in Feld **Pattern** direkt zu ersetzen (Schaltfläche **replace**). Dabei können Sie verschiedene Regex-Syntaxoptionen aktivieren (z.B. `CASE_INSENSITIVE` oder `UNICODE_CASE`).



URLCodec Hier können Sie eine aus der Zwischenablage eingefügte Zeichenkette URL-kodieren oder dekodieren





5.11 URL-Protokolle

Erfahren Sie, welche X4-spezifische URL-Protokolle verfügbar sind, mit denen Sie in XSL-Mappings u. a. auf Prozess-Variablen und Repository-Dateien zur Prozess-Laufzeit zugreifen können.

5.11.1 context:-URL-Protokoll

Mit dem `context:`-URL-Protokoll können Sie zur Laufzeit eines Prozesses in XSL-Mappings auf Variablen zugreifen, die im Prozess über den Variable-Baustein definiert und vor dem XSL-Mapping-Baustein im Prozess angeordnet wurden. Das in der Variable gespeicherte Dokument kann im XSL-Mapping über die `document()`-Funktion eingebunden werden.

Aufbau der URL

Die URL sollte nach folgendem Muster aufgebaut sein (Schrägstriche nach dem URL-Handler können ggf. problematisch sein):

```
context:Variablenname
```

> ⓘ Ein XSL-Mapping, das dieses URL-Protokoll verwendet, kann nicht in der Transformations-Vorschau des Mapping Editors ausgeführt werden.

Beispiel

In diesem Beispiel wurde in einem Prozess, der auch das folgende XSL-Mapping enthält, eine Variable *`Var`* über einen `Variable`-Baustein angelegt. Diese Variable speichert ein XML-Dokument, das im XSL-Mapping über die `document()`-Funktion gelesen und in einer XSL-Variable *`DocVar`* zwischengespeichert wird. Deren Knoten sind per XPath beliebig selektierbar.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:output encoding="UTF-8" method="xml" version="1.0"/>
   <xsl:template match="/">
      <result>
         <xsl:variable select="document('context:Var')" name="DocVar"/>
         <xsl:value-of select="$DocVar/child::ElemName"/>
      </result>
   </xsl:template>
</xsl:stylesheet>
```

5.11.2 xstore://-URL-Protokoll

Mit dem `xstore://`-URL-Protokoll können Sie zur Laufzeit eines Prozesses auf Dokumente im X4 Repository des aktuellen Benutzers zugreifen. In XSL-Mappings und in Adapter-Parametern können Sie damit beliebige Repository-Dateien referenzieren. XML-Dokumente können in XSL-Mappings über die `document()`-Funktion eingebunden werden.

Aufbau der URL

Eine `xstore://`-URL muss nach folgendem Muster aufgebaut sein:

```
xstore://<Dateipfad>
```

Bestandteile einer `xstore://`-URL:

- `xstore://`: Protokoll-Angabe; in der X4 BPMS zur Laufzeit des jeweils ausgeführten Prozesses gültig.
- `Dateipfad`: Repository-Pfad zur Datei im Benutzerordner; Ordner werden durch Schrägstriche (`/`) getrennt.

> Beachten Sie:
>
> - Um Dokumente aus einem anderen *X4 Repository*-Benutzerordner zu lesen, das `x4db:/`-Protokoll verwenden; siehe x4db:/-URL-Protokoll.
> - Ein XSL-Mapping, das dieses URL-Protokoll verwendet, kann nicht in der Transformations-Vorschau des Mapping Editors ausgeführt werden.

Beispiel
Im folgenden Beispiel wird in einem XSL-Mapping ein XML-Dokument im Repository über die `document()`-Funktion aufgerufen und in einer XSL-Variable *`externalFile`* zwischengespeichert. Die Knoten des Variableninhalts sind per XPath beliebig selektierbar.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:output encoding="UTF-8" method="xml" version="1.0"/>
   <xsl:template match="/">
      <result>
         <xsl:variable select="document('xstore://Project/Folder/Data.xml')"
name="externalFile"/>
         <xsl:value-of select="$externalFile/child::ElemName"/>
      </result>
   </xsl:template>
</xsl:stylesheet>
```

5.11.3 x4db:/-URL-Protokoll

Mit dem `x4db:/`-URL-Protokoll können Sie zur Laufzeit eines Prozesses auf Dokumente in einem beliebigen *X4 Repository*-Benutzerordner zugreifen. In XSL-Mappings und in Adapter-Parametern können Sie damit beliebige Repository-Dateien referenzieren. XML-Dokumente können in XSL-Mappings über die `document()`-Funktion eingebunden werden.

Aufbau der URL

Eine `x4db:/`-URL muss nach folgendem Muster aufgebaut sein:

```
x4db:/<Benutzer-Repository>/<Dateipfad>
```

Bestandteile einer `x4db:/`-URL:

- `x4db:/`: Protokoll-Angabe; in der X4 BPMS zur Laufzeit des jeweils ausgeführten Prozesses gültig.
- `Benutzer-Repository`: Repository-Benutzerordner (*"User collection"*), z.B. *`1`* oder *`X4P`*
- `Dateipfad`: Repository-Pfad zur Datei im Benutzerordner; Ordner werden durch Schrägstriche (`/`) getrennt.

> Beachten Sie:
>
> - Verwenden Sie das `x4db:/`-URL-Protokoll aus Sicherheitsgründen sparsam und ausschließlich zum Lesezugriff auf ein Dokument in einem anderen *X4 Repository*-Benutzerordner. Verwenden Sie ansonsten das `xstore://`-Protokoll; siehe „xstore://-URL-Protokoll.
> - Ein XSL-Mapping, das dieses URL-Protokoll verwendet, kann nicht in der Transformations-Vorschau des Mapping Editors ausgeführt werden.

Beispiel
Im folgenden Beispiel wird in einem XSL-Mapping ein XML-Dokument aus dem *X4 Repository*-Benutzerordner `101` über die `document()`-Funktion aufgerufen und in einer XSL-Variable

externalFile zwischengespeichert. Die Knoten des Variableninhalts sind per XPath beliebig selektierbar.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:output encoding="UTF-8" method="xml" version="1.0"/>
   <xsl:template match="/">
      <result>
         <xsl:variable select="document('x4db:/101/Project/Folder/Data.xml')"
name="externalFile"/>
         <xsl:value-of select="$externalFile/child::ElemName"/>
      </result>
   </xsl:template>
</xsl:stylesheet>
```

5.11.4 cryptstore://-URL-Protokoll

Mit dem `cryptstore://`-URL-Protokoll können Sie zur Laufzeit eines Prozesses in XSL-Mappings auf verschlüsselte Repository-Dokumente zugreifen und diese in einem Schritt entschlüsseln.
Das Dokument kann im XSL-Mapping über die `document()`-Funktion eingebunden werden, die den direkten Zugriff auf das dekodierte Dokument ermöglicht.

Verwenden einer Cryptstore-Implementierung

Der *X4 Server* ermöglicht über das `cryptstore://`-URL-Protokoll die Ver- und Entschlüsselung von Repository-Elementen. Dabei wird der Bytestrom durch eine kundenspezifische Java-Klasse geleitet, um gelesen, entschlüsselt und weitergegeben zu werden. Diese kundenspezifische Klasse muss von `InputStream` ableiten und einen Konstruktor besitzen, der einen `InputStream` als einzigen Parameter entgegennimmt.

Empfohlen wird, die Klasse von `javax.crypto.CipherInputStream` abzuleiten und im Konstruktor das Objekt zu parametrieren. Die fertige Klasse muss als Java-Archiv (`.jar`) im Ordner `<WildFly>\modules\system\layers\base\de\softproject\x4\client\main` angelegt werden. Zusätzlich muss in der Datei `module.xml` innerhalb von `<WildFly>\modules\system\layers\base\de\softproject\x4\client\main` im Element `<resources>` ein Kindelement `<resource-root path="<Name des Java-Archivs>.jar"/>` hinzugefügt werden. Darüber hinaus muss der Klassenname in der Konfigurationsdatei `<X4>/X4config.xml` direkt hinter dem Element `<xstore/>` in einem Element `<cryptstoreStreamHandlerClass/>` als vollqualifizierter Klassenname angegeben werden.

Aufbau der URL

Die URL muss nach folgendem Muster aufgebaut sein:

```
cryptstore://Repository-Pfad/VerschlüsselteDatei
```

Ein XSL-Mapping, das dieses URL-Protokoll verwendet, kann nicht in der Transformations-Vorschau des Mapping Editors ausgeführt werden.

Beispiel

Im folgenden Beispiel wird ein verschlüsseltes XML-Dokument im Repository innerhalb eines XSL-Mappings über die `document()`-Funktion aufgerufen und in einer XSL-Variable *`externalFile`* zwisch engespeichert. Die Knoten des Variableninhalts sind per XPath beliebig selektierbar.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:output encoding="UTF-8" method="xml" version="1.0"/>
   <xsl:template match="/">
      <result>
         <xsl:variable select="document('cryptstore://Project/Folder/Data.xml')"
 name="externalFile"/>
         <xsl:value-of select="$externalFile/child::ElemName"/>
      </result>
   </xsl:template>
</xsl:stylesheet>
```

Beispiel für eine mögliche Implementierung

```
import java.io.InputStream;
import java.security.InvalidKeyException;
import java.security.Key;
import java.security.NoSuchAlgorithmException;

import javax.crypto.Cipher;
import javax.crypto.CipherInputStream;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.spec.SecretKeySpec;

public class TestCrypt extends CipherInputStream {

    public TestCrypt(InputStream stream) throws InvalidKeyException,
NoSuchAlgorithmException, NoSuchPaddingException {
        super(stream,createCipher());
    }

    private static Cipher createCipher() throws InvalidKeyException,
NoSuchAlgorithmException, NoSuchPaddingException {
        // Ciphers must NOT be cached!!
        Cipher cipher = Cipher.getInstance( "AES" );
        Key key = new SecretKeySpec( readKeyBytes(), "AES" );
        cipher.init( Cipher.DECRYPT_MODE, key );

        return cipher;
    }

    private static byte[] readKeyBytes() {
        // read your Key from where ever it is needed
        // the keyData may be cached
        byte[] keyData;
        return keyData;
    }
}
```

5.11.5 globalcontext:-URL-Protokoll

Mit dem `globalcontext:`-URL-Protokoll können Sie zur Laufzeit eines Technical Processes in XSL-Mappings auf Kontextschlüssel zugreifen, die bereits über einen *Global Context Manipulation* Adapter definiert wurden. Das im Kontext gespeicherte Dokument kann im XSL-Mapping über die `document()`-Funktion eingebunden werden.

Globaler Kontext

Mit einem global definierten Kontext bietet die X4 BPMS die Möglichkeit, ähnlich wie bei Prozess-Variablen beliebige Dokumente aus einem Technical Process in einer globalen Hash-Tabelle zwischenzuspeichern, sodass auch von anderen Technical Processes auf diesen Kontext zugegriffen werden kann; siehe *Handbuch X4 Designer - Variablen-Baustein*. Die im globalen Kontext abgelegten Dokumente werden unter einem Schlüssel gespeichert und sind über die gesamte Laufzeit des X4 Servers verfügbar.

Aufbau der URL

Die URL muss nach folgendem Muster aufgebaut sein:

```
globalcontext:Schlüsselname
```

> Ein XSL-Mapping, das dieses URL-Protokoll verwendet, kann nicht in der Transformations-Vorschau des Mapping Editors ausgeführt werden.

Beispiel

In diesem Beispiel wurde in einem Technical Process, der auch das folgende XSL-Mapping enthält, ein Kontextschlüssel *X4Experience.GlobalContext.Sample* über einen *Global Context Manipulation* Adapter angelegt. Dieser Kontexteintrag speichert ein XML-Dokument, das im XSL-Mapping über die `document()`-Funktion gelesen und in einer XSL-Variable *DocVar* zwischengespeichert wird. Deren Knoten sind per XPath beliebig selektierbar.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:output encoding="UTF-8" method="xml" version="1.0"/>
   <xsl:template match="/">
      <result>
         <xsl:variable
          select="document('globalcontext:X4Experience.GlobalContext.Sample')"
name="DocVar"/>
         <xsl:value-of select="$DocVar/child::Data"/>
      </result>
   </xsl:template>
</xsl:stylesheet>
```

5.12 Logging von Technical Processes

In Technical Processes können Sie technische und fachliche Informationen protokollieren und Log-Nachrichten an die Monitoring-Komponente oder über Log4j z. B. in das Server-Log ausgeben.

Funktions-Adapter

Zum Logging von Technical Processes stehen Ihnen verschiedene Adapter zur Verfügung.

- `Process Monitoring`: Listet alle Prozessschritte und Ausführungszeiten eines Technical Processes auf und gibt detaillierte Informationen zu einem bestimmten Technical Process aus.
- `Process Monitoring Management`: Aktualisiert oder listet Profilings auf, die sich im X4 Repository oder in einem Projekt befinden.
- `Process Runtime Management`: Stoppt, beendet und listet laufende Technical Processes auf.
- `Log4J Logging`: Gibt Daten aus dem Prozess über *Log4J* in das Server-Log aus. Als Input-Daten können alle Daten dienen, die sich in eine Zeichenkette umwandeln lassen. Im `Advanced`

`Mode` können in einem XML-Dokument mit adapterspezifischer Struktur Daten auf verschiedenen Log-Stufen direkt ausgegeben werden.

Logging-Aktion auf Transitions

- Im X4 Designer können Sie eine Protokollierung einrichten, indem Sie zu Übergangslinien eine Logging-Aktion definieren. In der Sicht **Properties** können Sie für markierte Übergangslinien die Eigenschaft `Log4j Log` setzen und so das Logging-Verhalten konfigurieren.

.log-Datei

- Die `.log`-Datei wird beim Serverstart eingelesen.
- Der Dateiname der `.log`-Datei darf nicht geändert werden.

- Der X4 Server kann zu loggende Daten in eine `.log`-Datei im Serververzeichnis schreiben. Um die `.log`-Datei zu aktivieren, muss die `standalone.xml` des X4 Servers angepasst werden.

Standardmäßig werden die zu loggenden Daten in den Arbeitsspeicher geschrieben. Die Daten enthalten die Prozesse mit Prozesslaufzeiten und Prozess-Schrittlaufzeiten. Zusätzlich können Sie die X4 BPMS so konfigurieren, dass die Daten in eine .log-Datei oder in Elasticsearch geschrieben werden.

Die geloggten Daten können mit dem Adapter Process Monitoring ausgelesen und anschließend verwendet werden. Jedoch ist nur eine Quelle zum Lesen der Daten zulässig.

5.12.1 .log-Datei aktivieren

- Die `.log`-Datei wird beim Serverstart eingelesen.
- Der Dateiname der `.log`-Datei darf nicht geändert werden.

Um die .log-Datei zu aktivieren, sind folgende Änderungen in `<WildFly>\standalone\configuration\standalone.xml` erforderlich.

1. Fügen Sie im Element `<subsystem xmlns="urn:jboss:domain:logging:8.0">` folgenden Abschnitt hinzu:

```
<custom-handler name="PROFILING" class="org.jboss.logmanager.handlers.PeriodicS
izeRotatingFileHandler" module="org.jboss.logmanager">
   <formatter>
    <named-formatter name="PROFILING-PATTERN"/>
  </formatter>
    <properties>
      <property name="autoFlush" value="true" />
      <property name="append" value="true" />
      <property name="rotateSize" value="20000000" />
      <property name="maxBackupIndex" value="1" />
      <property name="suffix" value=".yyyy-MM-dd" />
      <property name="fileName" value="${jboss.server.log.dir}/profiling.log" /
>
  </properties>
</custom-handler>
```

2. Fügen Sie im Element `<logger category="X4-Profiling-Filesystem" use-parent-handlers="false">` folgenden Abschnitt hinzu:

```
<handlers>
  <handler name="PROFILING"/>
</handlers>
```

3. standalone.xml speichern.
4. X4 Server neu starten.

5.13 Save Points für wiederaufsetzbare Prozesse verwenden

Mit den sogenannten *Save Points* können Sie in der X4 BPMS den kompletten Zustand der Prozess-Engine mit den verarbeiteten Daten und Statusinformationen bei der Ausführung einer Technical Process-Instanz im Dateisystem persistieren.
Wird der Technical Process von einem Save Point fortgesetzt, werden die persistierten Daten verwendet. Werden in der weiteren Ausführung des Technical Processes Daten, Adapter und Subprozesse erstmals aufgerufen, dann werden diese in ihrer aktuellen Version eingebunden.

Save Points können Sie im X4 Designer definieren, indem Sie im Prozessdiagramm eine Übergangslinie (*Transition*) markieren und in der **Properties**-Sicht für die Eigenschaft `Save Point` eine der folgenden Operationen festlegen:

	`Save and Continue`	Save Point setzen und die Prozess-Ausführung direkt fortsetzen; Bei komplexeren Technical Processes können Sie damit Zwischenstände speichern. Falls Fehler oder Ausfälle auftreten, kann der Technical Process direkt ab dem letzten Save Point wiederaufgesetzt werden und muss nicht komplett neu ausgeführt werden.

	`Save and Stop`	Save Point setzen, dann die Ausführung der Technical Process-Instanz anhalten; Damit können Sie langlaufende Prozesse implementieren, die während der Ausführung "schlafen gelegt" werden können, bis beispielsweise Datenmanipulationen oder Wartungsarbeiten abgeschlossen sind. Anschließend können Sie die Technical Process-Instanz fortsetzen.
	`Delete`	aktuellen Save Point löschen

Einschränkungen

- Datenbank-Transaktionen oder Cursors lassen sich nicht persistieren.
- Zu jeder Prozess-Instanz lässt sich ein Save Point speichern. Werden innerhalb eines Prozesses mehrere Save Points gesetzt, so wird der jeweils letzte Stand des Save Points verwendet.
- Im Debug-Modus können Save Points nicht verwendet werden.

Ausführung einer Prozess-Instanz über die Monitoring-Oberfläche fortsetzen

1. Die Monitoring-Oberfläche im Browser aufrufen.
2. Den entsprechenden angehaltenen Prozess in der Instanzsicht wählen.
   Der Prozess wird nun in der Detailansicht angezeigt.
3. Unter **Savepoints** den entsprechenden Save Point wählen.
4. **Neustart/Fortsetzen** klicken, um die entsprechende Prozess-Instanz fortzusetzen.
   Die Ausführung der Prozess-Instanz wird nun auf Basis des aktuellsten Save Points mit allen verarbeiteten Daten und dem persistierten Zustand der Prozess-Engine fortgesetzt.

5.14 Webservices via HTTP(S) bereitstellen

Mit der zentralen Webservice-Schnittstelle lassen sich Technical Processes als Webservices via HTTP(S) bereitstellen, z. B. für ReSTful-Webservices oder SOAP-Webservices. Die einzelnen Ressourcen eines Webservices lassen sich dabei von außen über einen HTTP-URL (Uniform Resource Locator) ansprechen.

Basis-URL und Funktionsweise

Die Basis-URL der zentralen Webservice-Schnittstelle lautet `http://<server>/X4/httpstarter/ReST` (z. B. `http://localhost:8080/X4/httpstarter/ReST`).

HTTP-Anfragen werden von der zentralen Webservice-Schnittstelle unmittelbar verarbeitet und als XML-Dokument an einen Prozess übergeben. Das Ergebnis des Prozesses, z. B. ein erzeugtes Dokument, kann direkt zurückgegeben werden.

Konfiguration

Webservices werden im X4 Designer über den grafischen *Webservice Configuration Editor* konfiguriert.

Rückgabewerte

Als Rückgabewert liefert die Webservice-Schnittstelle den aktuellen Statuscode (z. B. `201 Created`) und je nach Anfrage die identifizierte Ressource.

Unterstützte Webservice-Typen

- **ReSTful-Webservices**: *Representational State Transfer* (ReST) ist ein Architekturstil, bei dem sich jede Ressource eines Informationssystems unter einem eigenen URI z. B. über HTTP ansprechen lässt. Ziel von ReSTful Services im X4 Server ist dementsprechend, Pfadstrukturen eines URI nachvollziehbar und transparent auf Prozesse abzubilden.
  Für ReSTful-Webservices lassen sich pro Ressource mehrere Operationen bzw. HTTP-Methoden zuordnen und mit jeweils einem Prozess verknüpfen, z. B. für Zugriffe per `GET`, `POST` oder `DELETE`. Hierbei werden folgende HTTP-Methoden unterstützt:
  - `GET`
  - `PUT`
  - `POST`
  - `DELETE`
  - sowie beliebige weitere Methoden, die aus gültigen XML-Zeichen bestehen.

  Beim Webservice-Aufruf wird standardmäßig eine bestimmte XML-Datenstruktur an den X4-Prozess übergeben, siehe ReSTful-Webservices: Input-XML-Struktur. Zudem erwartet die Webservice-Schnittstelle ein XML-Dokument als Ergebnis, das gemäß einer definierten Struktur aufgebaut ist und eine Reihe von Optionen bietet.
  Die übertragenen Daten müssen in Base64 oder Base16 (hexadezimal) kodiert werden, da über HTTP nur Bytes übertragen werden. Um Probleme bei der Interpretation des Inhalts im Element `<Body>` zu vermeiden, wird dort direkter Text oder XML unterstützt. Der Text bzw. das XML-Dokument muss also zunächst in Bytes umgewandelt und Base64/Base16-kodiert werden, damit Sie die volle Kontrolle über die Bytes besitzen, die als Antwort zurückgegeben werden.

  > Weiterführende Informationen zum ReST-Architekturstil finden Sie in der englischsprachigen Dissertation von Roy T. Fielding unter http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.

- **SOAP-Webservices**: SOAP ist ein vom W3C standardisiertes Nachrichtenprotokoll (http://www.w3.org/TR/soap), welches eine standardisierte Kommunikation zwischen verteilten Anwendungen ermöglicht. SOAP-Webservices können in verschiedenen Varianten bereitgestellt werden, u.a. mit externer WSDL-Definition oder ohne WSDL-Definition, im RPC-Stil oder im Document-Modus. Als Nachrichtenformat wird bei SOAP-Webservices typischerweise XML für Anfrage (Request) und Ergebnis (Response) genutzt, und Binärdaten werden ggf. per MIME anhängt (MTOM).
  SOAP-Webservices lassen sich auf die gleiche Weise wie ReSTful-Services via HTTP(S) bereitstellen, siehe SOAP-Webservices bereitstellen.
  Der Service-Typ SOAP unterstützt die Standards SOAP 1.1 und SOAP 1.2. SOAP-Service-

Definitionen können beliebig mit HTTP-ReST-Webservice-Definitionen und mit File-Webservice-Definitionen kombiniert werden.

- **File-Webservices**: Mit der zentralen Webservice-Schnittstelle lassen sich auch statische Datei-Ressourcen aus dem Repository bzw. dynamisch per X4-Prozess erzeugte Dokumente als Service via HTTP-GET bereitstellen, siehe File-Webservice bereitstellen.

5.14.1 Webservice-Konfiguration grafisch bearbeiten

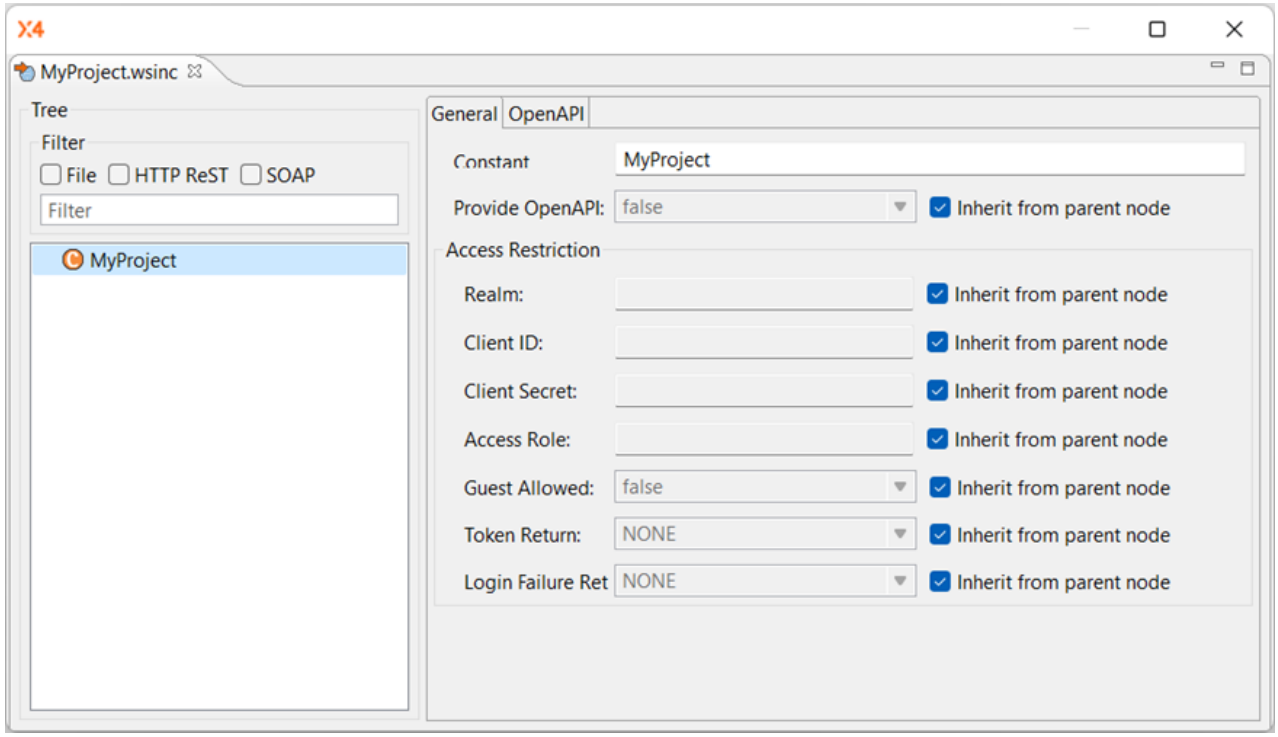
Die Webservice-Konfiguration lässt sich im *X4 Designer* über den grafischen *Webservice Configuration Editor* bearbeiten und im *X4 Server* aktualisieren. Technical Processes können so mit wenigen Mausklicks als HTTP-basierte Webservices verschiedener Ausprägungen (ReST, SOAP etc.) bereitgestellt werden.
Alle Webservices, die auf diesem Weg bereitgestellt werden, besitzen eine einheitliche Basis-URL. Die Basis-URL ist standardmäßig http://localhost:8080/X4/httpstarter/ReST.

5.14.1.1 Webservice Configuration Editor aufrufen

Um den *Webservice Configuration Editor* aufzurufen, doppelklicken Sie in der Sicht **Projects** auf die `.wsinc`-Datei.

5.14.1.2 Oberfläche des Webservice Configuration Editors

Der *Webservice Configuration Editor* ist in folgende Bereiche unterteilt:



5.14.1.2.1 Tree: URL-Baumstruktur

Im Bereich **Tree** wird die hierarchische Webservice-URL-Struktur bzw. werden die URL-Parts in einer auf- und zuklappbaren Baumstruktur angezeigt. Diese Baumstruktur können Sie per Rechtsklick über das Kontextmenü bearbeiten. Zudem können Sie Elemente innerhalb des Baums per Drag&Drop verschieben.

Über das Feld **Filter** können Sie die angezeigte Baumstruktur anhand des Knotennamens filtern, sodass Strukturen, die das Suchwort nicht enthalten, ausgeblendet werden. Über ✖ oder durch Leeren des Feldes können Sie den Filter zurücksetzen. Zudem kann hier mit `File`, `HTTP ReST` und `SOAP` auch anhand des Service-Typs gefiltert werden.

Baumelemente

Constant	Konstanter URL-Part Sämtliche Eigenschaften der ausgewählten Konstante können Sie im Bereich **Scope** definieren oder Sicherheitseigenschaften vom übergeordneten Element erben lassen.
Variable	Variabler URL-Part Sämtliche Eigenschaften zur ausgewählten Variable können Sie im Bereich **Scope** definieren oder Sicherheitseigenschaften vom übergeordneten Element erben lassen.
Remainder	Komplett beliebige URL-Part-Definition Sämtliche Eigenschaften zum ausgewählten Remainder können Sie im Bereich **Scope** definieren oder Sicherheitseigenschaften vom übergeordneten Element erben lassen.

5.14.1.3 Konstanten, Variablen oder Remainder erstellen und verwalten

Markieren Sie in der Baumstruktur den gewünschten Knoten (entspricht einem URL-Part des Webservices), klicken Sie mit der rechten Maustaste darauf, und wählen Sie im Kontextmenü eine Aktion:

Add Constant	Konstante als Kindknoten anlegen, um einen konstanten URL-Part für Ihren Webservice zu definieren
Add Variable	Variable als Kindknoten anlegen, um mit Platzhaltern variable Bestandteile innerhalb der URL Ihres Webservices zu definieren
Add Remainder	Remainder als Kindknoten anlegen, um alle restlichen URL-Parts abzufangen
Copy Service URL to Clipboard	Zugehörige Webservice-Ressourcen-URL in die Zwischenablage kopieren

Rename	Markierten Knoten umbenennen
Delete	Markierten Knoten löschen

Bei den Knotennamen wird zwischen Groß- und Kleinschreibung unterschieden!

5.14.1.4 Sicherheitseigenschaften des Knotens definieren

Für den ausgewählten Knoten können Sie in **Access Restrictions** bei Bedarf (Sicherheits-)Eigenschaften definieren.

Mit **Inherit from parent node** wird der Sicherheitsbereich vom übergeordneten Element übernommen. Deaktivieren Sie diese Eigenschaft, um einen benutzerdefinierten Wert zu setzen.

Realm	Sicherheitsbereich für den aktuellen URL-Part
Client ID	Keycloak-Client
Client Secret	Keycloak-Client-Anmeldedaten
Access Role	Keycloak-Role Die Role muss in Keycloak dem Benutzer zugewiesen werden, um Zugang zum Webservice zu erhalten.
Guest Allowed	Wenn ein Sicherheitsbereich in Eigenschaft `Realm` definiert wurde, kann mit *true* ein Gastzugriff erlaubt werden
Token Return	Zurückgegebenes Log-in-Token auswählen **Mögliche Werte:** • *NONE* (Standard) • *CUSTOM_HEADER*

Login Failure Return	In Kombination mit dem Log-in-Modul für *X4 Web Apps* können Sie mit **Login Failure Return** steuern, ob nach einem gescheiterten Anmeldeversuch auch der Grund zurückgegeben werden soll. **Mögliche Werte:** • *NONE* (Standard): keine Information über den Grund des Scheiterns zurückmelden (wird im Server-Log protokolliert) • *HEADER*: den technischen oder fachlichen Grund im HTTP-Header `X-X4-Login-Failed` ausgeben.

5.14.2 ReSTful-Webservices bereitstellen

Für jeden URL-Part Ihres ReSTful-Webservices können Sie pro HTTP-Methode jeweils einen verarbeitenden Technical Process zuordnen, z. B. für die *GET*-Methode einen Eintrag `GET`.

Verfügbare HTTP-Methoden

Innerhalb einer Operations-Zuordnung können folgende HTTP-Methoden jeweils einmal definiert werden:

- `GET`
- `PUT`
- `POST`
- `DELETE`
- sowie beliebige weitere Methoden, die einen gültigen XML-Namen besitzen

5.14.2.1 Technical Process als ReSTful HTTP-Webservice bereitstellen

Um im Webservice Configuration Editor einen Technical Process als Operation eines ReSTful-Webservices bereitzustellen, gehen Sie wie folgt vor:

1. Legen Sie in der Baumstruktur bei Bedarf einen neuen Knoten an (entspricht einem URL-Part des Webservices). Weitere Informationen hierzu finden Sie unter Webservice-Konfiguration grafisch bearbeiten.
2. Legen Sie für den gewünschten Knoten (vom Typ `Constant`, `Variable` oder `Remainder`) über **Add HTTP ReST Service** einen neuen Service vom Typ *HTTP ReST* an. Passen Sie ggf. die (Sicherheits-)Eigenschaften an.
3. Legen Sie für den HTTP-ReST-Service über das Kontextmenü **Add HTTP ReST Operation** eine neue Operation an, oder wählen Sie in der Tabelle **Operations** einen bestehenden Eintrag aus, der überschrieben werden soll.

> Sie können jeweils einmal pro Service eine Operation `GET`, `POST`, `PUT`, `DELETE` vergeben oder einen frei wählbaren Operationsnamen eingeben Bei den Operationsnamen wird zwischen Groß-/Kleinschreibung unterschieden.

4. Ziehen Sie einen Prozess aus der Ansicht **Projects** per Drag&Drop auf die gewünschte Operation, um den Prozesspfad zuzuordnen.
5. Definieren Sie Input-Modus und Output-Modus für jede Operation, um festzulegen welche XML-Struktur erwartet und erzeugt wird.
6. Aktivieren Sie ggf. die Nutzung von Platzhaltern, um Parameter für die Webservice-Operation zu verarbeiten und Informationen aus dem HTTP-Request als Platzhalter bereitzustellen.

> Folgende Platzhalter mit Informationen aus dem HTTP-Request stehen bei Option `Enabled` innerhalb des X4-Prozesskontexts zur Verfügung, z. B. innerhalb von XSL-Mappings:
>
> - `REQUEST_PARAMETER_<`*Name*`>`: Gewünschter Parameter (Groß-/Kleinschreibung wird beachtet!)
> - `REQUEST_HEADER_<`*NAME*`>`: Gewünschter HTTP-Header (nur Großschreibung!)
> - `REQUEST_METHOD`
> - `REQUEST_SCHEME`
> - `REQUEST_PATH`
> - `REQUEST_FULL_PATH`
> - `REQUEST_REMOTE_ADDRESS`
> - `REQUEST_REMOTE_PORT`
> - `REQUEST_REMOTE_USER`
> - `REQUEST_LOCAL_ADDRESS`
> - `REQUEST_LOCAL_PORT`
> - `REQUEST_SERVER_NAME`
> - `REQUEST_SERVER_PORT`

7. Definieren Sie Sicherheits-Optionen wie `Realm` oder `Guest Allowed` für die Operation.

Beispiel: ReSTful Webservice-Konfiguration (Auszug XML-Speicherformat)

```
<Constant value="example">
    <Service type="Http">
        <Operation name="GET" process="x4db:/1/RestExamples/http/DirectOutput.wrf">
            <InputHandler type="HTTP-MetaXml" />
            <OutputHandler type="HTTP-Content" />
        </Operation>
    </Service>
</Constant>
```

5.14.2.2 ReSTful-Webservices: Input-XML-Struktur

Beim Aufruf eines Webservices wird ein Dokument an den entsprechenden Technical Process übergeben. Wenn Sie in der Webservice-Konfiguration als Input-Modus `HTTP-Content` definiert haben, handelt sich dabei ausschließlich um den Inhalt der HTTP-Entität. `Media Type` und `Charset` a us dem Header `Content-Type` werden auf das Input-Dokument übertragen.
Wenn als Input-Modus `HTTP-MetaXml` definiert wurde, wird beim Webservice-Aufruf ein XML-Dokument u.a. mit HTTP-Header-Informationen an den Technical Process übergeben, das nach folgendem Muster aufgebaut ist:

```
<Request>
  <Path>
    <Method>ReST-Methode (z. B. GET)</Method>
    <FullPath>Komplette URL inkl. konstantem und variablem URL-Part</FullPath>
    <Server>URL des Servers</Server>
    <RequestPath>Pfad, der im Aufruf verwendet wurde</RequestPath>
    <ContextPrefix>Basis-URL von X4 ReST (immer gleich)</ContextPrefix>
    <PathInfo>Konstanter und variabler URL-Part</PathInfo>
    <Parts>
      <Constant name="Konstanter URL-Part" />
      <Variable name="Name des variablen URL-Parts">tatsächl. URL-Part</Variable>
    </Parts>
  </Path>
  <Headers>
    <Header>
      <Name>Header-Name</Name>
      <Value>Header-Wert</Value>
   </Header>
   <!-- Oder bei Datumswerten: -->
   <Header>
      <Name>HTTP-Header-Name</Name>
      <Date timezone="Zeitzone">Header-Datumswert</Date>
   </Header>
  </Headers>
  <Query>
    <QueryString>Über die URL übergebene Parameter</QueryString>
    <Parameter name="Parameter-Name">Parameter-Wert</Parameter>
  </Query>
  <Security realm="Sicherheitsbereich">
     <Principal class="Prinzipalklasse" name="Prinzipal-Name" />
  </Security>
  <Body encoding="Datenkodierung">Übermittelte Daten</Body>
</Request>
```

Das Wurzel-Element `<Request>` enthält folgende Elemente:

- `<Path>`: Enthält u. a. die aufrufende URL und die einzelnen aufgeschlüsselten URL-Parts
- `<Headers>`: Enthält in jeweils einem Element `<Header>` die einzelnen HTTP-Header mit Name und Wert.

- <Query>: Enthält die Elemente <QueryString> und ggf. für jeden Parameter ein Element <Parameter>
- <Security>: Enthält die zugeordneten JAAS-Subjects (einfache Prinzipale oder Prinzipal-Gruppen), falls es sich um keinen Gastzugriff handelt
- <Body>: Enthält in Attribut encoding den Kodierungstyp *base64* und als Elementinhalt die entsprechend kodierten Nutzdaten

5.14.2.3 ReSTful Webservices: Erwartete Output-XML-Struktur

Wenn für die jeweilige URI-Operation in der Webservice-Konfigurationsdatei der Output-Handler HTTP-MetaXml definiert wurde, erwartet die Webservice-Schnittstelle als Ergebnis des verarbeitenden Prozesses ein XML-Dokument, das nach folgendem Muster aufgebaut ist:

```
<Response>
   <Status code="HTTP-Status-Code"/>
   <Headers>
      <Header>
         <Name>HTTP-Header-Name</Name>
         <Value>Header-Wert</Value>
      </Header>
      <!-- Oder bei Datumswerten: -->
      <Header>
         <Name>HTTP-Header-Name</Name>
         <Date timezone="Zeitzone">Header-Datumswert</Date>
      </Header>
   </Headers>
   <Content encoding="Kodierung">Übermittelte Daten</Content>
</Response>
```

Das Wurzel-Element <Response> enthält folgende Elemente:

- <Status>: Enthält in Attribut code eine dreistellige Zahl mit dem HTTP-Statuscode (z. B. *202*).
- <Headers>: Enthält beliebig viele <Header>-Elemente mit HTTP-Header-Informationen. Jedes <Header>-Element enthält einen HTTP-Header, der über die Kindelemente <Name> und <Value> oder <Date> (bei Datumsangaben im Format yyyy-MM-dd HH:mm:ss gemäß *SimpleDateFormat*)) definiert wird. Für Element <Date> kann im optionalen Attribut timezone die Zeitzone angegeben werden.
- <Content> (optional): Enthält in Attribut encoding den Kodierungstyp (*base64* oder *base16* bzw. *hex*) und als Elementinhalt die entsprechend kodierten Nutzdaten und/oder die Stream-Elemente.

```
<Content encoding="base64">
    ... Base64...
    <Stream url="http://remote.host/Some/Document?id=foo" transcoding="base64-
us-ascii"/>
    ... mehr Base64...
</Content>
```

5.14.3 SOAP-Webservices bereitstellen

Jeder SOAP Webservice wird über einen Knoten `<Services/>` innerhalb der Webservices-Konfiguration definiert. Dessen Attribute und Kindelemente variieren z.T. je nach Webservice-Typ.

Folgende Arten von SOAP Webservices stehen im Kindelement `<Wsdl>` in Attribut `mode` zur Auswahl:

- `RPC`: Es wird vom X4 Server eine WSDL-Definition nach RPC-Stil aus den Konfigurationsdaten erzeugt, siehe SOAP-Webservice im RPC-Stil bereitstellen
- `DOCUMENT`: Es wird vom X4 Server eine WSDL-Definition nach Document-Stil aus den Konfigurationsdaten erzeugt, SOAP-Webservice im Document-Modus bereitstellen
- `EXTERNAL`: Externe WSDL-Definition für den SOAP Webservice verwenden, siehe SOAP-Webservice mit externer WSDL-Definition bereitstellen
- `NONE`: Es wird keine WSDL-Definition verwendet/bereitgestellt, siehe SOAP-Webservice ohne WSDL-Definition bereitstellen
- `PROCESS`: Prozess wählen, der die WSDL erzeugt und als HTTP-Content an den Aufrufer zurückgibt. Das Ergebnis des Prozesses wird dabei 1:1 zurückgegeben.

Schematischer Aufbau einer SOAP Webservice-Konfiguration (XML-Speicherformat)

```
<Service type="Soap">
  <SoapVersion>
    <SOAP11/> <!-- für SOAP 1.1-Unterstützung -->
    <SOAP12/> <!-- für SOAP 1.2-Unterstützung -->
  </SoapVersion>
  <Wsdl mode="WSDL-Modus"/>
  <Namespace prefix="" namespace="" schemaLocation="URL zu einem XSD-Dokument"/>
  <Operation name="Operationsname" process="Pfad zum X4-Prozess als x4db:/-URL">
    <InputHandler type="Input-Modus"/>
    <OutputHandler type="Output-Modus"/>
    <Input>
      <Part name="Bezeichnung für einen Teil" type="Typ (optional)" element="Elementn
ame (optional)" />
    </Input>
    <Output>
      <Part name="Bezeichnung für einen Teil" type="Typ (optional)" element="Elementn
ame (optional)" />
    </Output>
  </Operation>
</Service>
```

Erläuterungen:

- Knoten `<Services type="Soap">` definiert einen HTTP-Webservice vom Typ SOAP.
- Knoten `<Wsdl>` definiert über dessen Attribut `mode` den gewünschten WSDL-Modus (z.B. `EXTERNAL`, siehe oben).
- Knoten `<SOAP11/>` und `<SOAP12/>` ermöglichen zu definieren, welche SOAP-Versionen der Webservice nutzt (entweder eine oder beide parallel).

- In beliebig vielen Knoten `<Namespace>` lassen sich jeweils ein Namespace für die Definition von Webservices verwenden.
- In beliebig vielen Knoten `<Operation>` lassen sich jeweils eine Operation für den Webservice definieren (siehe unten).

Folgende Namespaces werden vom System definiert:

- `xmlns:xsd=" http://www.w3.org/2001/XMLSchema "`
- `xmlns:soap11=" http://schemas.xmlsoap.org/wsdl/soap/ "`
- `xmlns:soap12=" http://schemas.xmlsoap.org/wsdl/soap12/ "`
- `xmlns:wsdl=" http://schemas.xmlsoap.org/wsdl/ "`

Request Routing für SOAP-Services

- **`GET`-Requests:** Ein `GET`-Request wird beim WSDL-Modus `NONE` mit dem HTTP-Status `405` (Meth od not allowed) beantwortet. Im WSDL-Modus `EXTERNAL` wird mit einem HTTP redirect auf die konfigurierte WSDL-URL geantwortet. Bei den WSDL-Modi `RPC` und `DOCUMENT` generiert der X4 Server aus der Service-Konfiguration eine WSDL als Antwort.
- **`POST`-Requests:** Bei einem `POST`-Request wird der komplette Request-Body in den Speicher geladen, damit das Routing auf die Operation erfolgen kann.
  a. Zunächst wird nach dem SOAP-Dokument gesucht.
     - Handelt es sich nicht um ein Multipart-Request, wird angenommen, dass der Request-Body das SOAP-Dokument enthält.
     - Handelt es sich um ein Multipart-Request (MTOM), wird nach dem Start-Part des MTOM-Pakets gesucht.
  b. Das gefundene Dokument wird als XML gelesen und die SOAP-Version aus dem Namespace des Wurzelelements ermittelt. Falls es sich dabei nicht um eine akzeptierte SOAP-Versionen handelt, wird mit einem entsprechenden SOAP-Fault geantwortet.
  c. Aus dem XML-Dokument wird der `Namespace` und `Localname` des ersten Kindelements des SOAP-Body herangezogen, um die Operation im Service zu ermitteln.

     > - Bei den WSDL-Modi `NONE`, `RPC` und `EXTERNAL` wird mit dem konfigurierten `Namespace` und `Localname` der Operation verglichen.
     > - Für den WSDL-Modus `DOCUMENT` wird die Konfiguration des jeweils ersten Input-Parts verwendet. Wenn es sich um einen *Element*-Part handelt (d.h. das Attribut `element` enthält einen Wert) so wird der *Qualifiedname* des konfigurierten Elementes verwendet. Für *Type*-Parts (d.h. das Attribut `type` enthält einen Wert) wird der Name des Parts (Attribut `name`) verwendet.

  d. Anschließend wird gemäß Input-Modus der Request aufbereitet und an den konfigurierten Prozess für die Operation übergeben.

     > Es können keine SOAP-Requests mit leerem `Body`-Element verarbeitet werden.

5.14.3.1 SOAP-Webservice im RPC-Stil bereitstellen

Wie Sie mit der X4 BPMS einen Technical Process als SOAP-Webservice im RPC-Stil bereitstellen, der zudem eine automatisch aus der Service-Konfiguration erzeugte WSDL-Definition bereitstellt, wird im Folgenden beschrieben.

Voraussetzungen:

- Grundlegendes Wissen über den Aufbau von SOAP-Webservices im RPC-Stil
- Ein Technical Process pro Webservice-Operation, der die Verarbeitung der eingehenden Anfrage übernimmt. Der Tehnical Process wird dabei über seine `x4db:/-URL` referenziert.
- Ein URL-Part, über den der SOAP-Webservice via HTTP erreichbar sein soll, z.B. über `http://localhost:8080/X4/httpstarter/ReST/`***`MyRPCWebservice`***. Diesen URL-Part haben Sie bereits typischerweise als Konstante innerhalb der Webservices-Konfiguration definiert.

Um einen Technical Process als SOAP-Webservice im RPC-Stil bereitzustellen, gehen Sie wie folgt vor:

1. Unter der gewünschten URL-Ressource (definiert z.B. durch einen konstanten URL-Part) über das Kontextmenü **Add SOAP Service** einen neuen Service-Eintrag vom Typ *SOAP* anlegen.
2. SOAP-Version festlegen (entweder *SOAP 1.1*, *SOAP 1.2* oder Unterstützung für beide Standards).
3. WSDL-Modus in **WSDL Mode** und Zielpräfix in **Service Prefix** wählen.
4. Für jede Operation folgende Konfiguration hinterlegen:
   - Einen gültigen Namen für die Operation hinterlegen.
     SOAP-Webservices im RPC-Stil erfordern eindeutige *Localnames* für die Operationen.
   - Technical Process, der die Webservice-Anfrage über den X4 Server verarbeitet, über dessen `x4db:/-URL` verknüpfen.
     Im *Webservice Configuration Editor* ist dies per Drag & Drop aus dem *Repository Navigator* möglich.
   - Input Handler und Output Handler für den SOAP-Webservice definieren.
     Beispiel: `SOAP-Content` definieren, um das erste Element im `soap:Body` als Input zu verwenden sowie beim Output ein `soap:Body`- und `soap:Envelope`-Element zu erzeugen.
   - Erwartete Input-XML-Datenstruktur innerhalb von Element `<Input>`angeben, optional mit XML-Schemadefinition.
   - Zu erzeugende Ausgabe-Datenstruktur innerhalb von Element `<Output>` angeben.
5. Speichern und Webservice-Konfiguration mit **Reload on Server** anwenden.
   Der SOAP-Webservice im RPC-Stil ist ab sofort aktiv.

Generierte WSDL-Definitionen im RPC

Die automatisch generierte zugehörige WSDL-Definition verwendet den RPC-Stil und wird bei einem HTTP-GET-Aufruf des SOAP-Webservice-URIs ausgeliefert.

Für den WSDL-Modus RPC erzeugt der X4 Server eine WSDL-Definition aus der Service-Konfiguration. Die WSDL-Definition wird wie folgt erstellt:

- Für jedes Namespace-Element mit schemaLocation-Attribut wird unter wsdl:types ein xsd:schema mit xsd:include für dieschemaLocation erzeugt.
- Für jede Operation wird eine wsdl:message erzeugt, deren Name dem *Localname* der konfigurierten Operation entspricht. Innerhalb dieser Message wird für jedes Input-Part ein entsprechendes wsdl:parterstellt.
- Für jede Operation wird eine wsdl:message erzeugt, deren Name dem *Localname* der konfigurierten Operation, ergänzt um Response, entspricht. Innerhalb dieser Nachricht wird für jedes *Output*-Part ein entsprechendes wsdl:part erstellt.
- Für jede Operation wird eine wsdl:operation und damit ein wsdl:portTyp erzeugt. Der Name der wsdl:operation ist der*Localname* der konfigurierten Operation. Als Input-/Output-Nachricht werden die oben erzeugten Nachrichten verwendet.
- Je nach SOAP-Version werden entsprechende wsdl:bindings erzeugt (ausschließlich literal Bindings). Es wird immer eine leere SOAP-Action definiert. Bei SOAP-Webservices mit RPC-Stil wird für jede Operation eintargetNamespace deklariert, falls der Operationsname ein *Qualifiedname* ist (d.h. ein Name mit Doppelpunkt).

Beispiel: Konfiguration eines SOAP-Webservices im RPC-Modus

```
<Service type="Soap">
    <SoapVersion>
        <SOAP11/>
        <SOAP12/>
    </SoapVersion>
    <Wsdl name="NameOfTheRpcService" targetPrefix="service" mode="RPC"/>
    <Namespace prefix="service" namespace="http://example.org/serviceSpace"/>
    <Namespace prefix="exampleSpace" namespace="http://example.org/namespace"
     schemaLocation="http://localhost:8080/X4/httpstarter/ReST/soapService/xsd/
namespace.xsd"/>
    <Namespace prefix="anotherSpace" namespace="http://example.org/different/
namespace"
     schemaLocation="http://localhost:8080/X4/httpstarter/ReST/soapService/xsd/
differentNamespace.xsd"/>
    <!-- RPC requires unique localnames for the operations -->
    <Operation name="exampleSpace:RpcOp1"
        process="x4db:/1/RestExamples/soap/DirectOutput.wrf">
        <InputHandler type="SOAP-Envelope"/>
        <OutputHandler type="SOAP-Envelope"/>
        <Input>
            <Part name="firstPart" element="exampleSpace:Operation1"/>
        </Input>
        <Output>
            <Part name="firstPart" element="exampleSpace:Operation1"/>
        </Output>
    </Operation>
    <Operation name="anotherSpace:RpcOp2"
        process="x4db:/1/RestExamples/soap/DirectOutput.wrf">
        <InputHandler type="SOAP-Envelope"/>
        <OutputHandler type="SOAP-Envelope"/>
        <Input>
            <Part name="firstPart" type="anotherSpace:SomeType"/>
        </Input>
        <Output>
            <Part name="Output" type="anotherSpace:SomeType"/>
        </Output>
    </Operation>
    <Operation name="RpcOp3"
        process="x4db:/1/RestExamples/soap/DirectOutput.wrf">
        <InputHandler type="SOAP-Envelope"/>
        <OutputHandler type="SOAP-Envelope"/>
        <Input>
            <Part name="Value" type="xsd:integer"/>
        </Input>
        <Output>
            <Part name="Value" type="xsd:String"/>
        </Output>
    </Operation>
</Service>
```

5.14.3.2 SOAP-Webservice im Document-Modus bereitstellen

Wie Sie mit der X4 BPMS einen SOAP-Webservice im Document-Stil bereitstellen, wird im Folgenden beschrieben.

Voraussetzungen:

- Grundlegendes Wissen über den Aufbau von SOAP-Webservices im Document-Stil, bei welchen der Request/Response Body per XML-Schema (XSD) vollständig beschrieben sind.
- Ein Technical Process pro Webservice-Operation, der die Verarbeitung der eingehenden Anfrage übernimmt. Der Technical Process wird dabei über seine `x4db:/-URL` referenziert.
- Ein URL-Part, über den der SOAP-Webservice via HTTP erreichbar sein soll, z.B. über `http://localhost:8080/X4/httpstarter/ReST/MyDocumentWebservice`. Diesen URL-Part haben Sie bereits typischerweise als Konstante innerhalb der Webservices-Konfiguration definiert, siehe Webservice-Konfiguration grafisch bearbeiten.

Um einen Technical Process als SOAP-Webservice im Document-Stil bereitzustellen, gehen Sie wie folgt vor:

1. Unter der gewünschten URL-Ressource (definiert z.B. durch einen konstanten URL-Part) über das Kontextmenü **Add SOAP Service** einen neuen Service-Eintrag vom Typ *SOAP* anlegen.
2. SOAP-Version festlegen (entweder *SOAP 1.1*, *SOAP 1.2* oder Unterstützung für beide Standards).
3. WSDL-Modus in **WSDL Mode** und Zielpräfix in **Service Prefix** wählen.
4. Für jede Operation folgende Informationen hinterlegen:
   - Einen gültigen Namen für die Operation hinterlegen.
     SOAP-Webservices im Document-Modus erfordern eindeutige *Localnames* für die Operationen.
   - Namespace-Deklaration für die Daten im Request/Response Body hinterlegen.
   - Technical Process, der die Webservice-Anfrage über den X4 Server verarbeitet, über dessen `x4db:/-URL` verknüpfen.
     Im *Webservice Configuration Editor* ist dies per Drag & Drop aus dem Repository Navigator möglich.
   - Input Handler und Output Handler für den SOAP-Webservice definieren.
     Beispiel: `SOAP-Content` definieren, um das erste Element im `soap:Body` als Input zu verwenden sowie beim Output ein `soap:Body`- und `soap:Envelope`-Element zu erzeugen.
   - Erwartete Input-XML-Datenstruktur innerhalb von Element `<Input>` inklusive Namespace angeben.
   - Zu erzeugende Ausgabe-Datenstruktur innerhalb von Element `<Output>` inklusive Namespace angeben.
5. Speichern und Webservice-Konfiguration mit **Reload on Server** anwenden.
   Der SOAP-Webservice im Document-Stil ist ab sofort aktiv.

Generierte WSDL-Definition im Document-Modus

Die automatisch generierte zugehörige WSDL-Definition verwendet den Document-Stil und wird bei einem `HTTP GET`-Aufruf der SOAP-Webservice-URLs ausgeliefert.

Für den WSDL-Modus `DOCUMENT` erzeugt der *X4 Server* eine WSDL-Definition aus der Service-Konfiguration. Die WSDL-Definition wird wie folgt erstellt:

- Für jedes Namespace-Element mit `schemaLocation`-Attribut wird unter `wsdl:types` ein `xsd:schema` mit `xsd:include` für die `schemaLocation` erzeugt.
- Für jede Operation wird eine `wsdl:message` erzeugt, deren Name dem *Localname* der konfigurierten Operation entspricht. Innerhalb dieser Message wird für jedes `Input`-Part ein entsprechendes `wsdl:part` erstellt.
- Für jede Operation wird eine `wsdl:message` erzeugt, deren Name dem *Localname* der konfigurierten Operation, ergänzt um `Response`, entspricht. Innerhalb dieser Nachricht wird für jedes *Output*-Part ein entsprechendes `wsdl:part` erstellt.
- Für jede Operation wird eine `wsdl:operation` und damit ein `wsdl:portTyp` erzeugt. Der Name der `wsdl:operation` ist der *Localname* der konfigurierten Operation. Als Input-/Output-Nachricht werden die oben erzeugten Nachrichten verwendet.
- Je nach SOAP-Version werden entsprechende `wsdl:bindings` erzeugt (ausschließlich `literal` Bindings). Es wird immer eine leere SOAP-Action definiert.

Beispiel: Konfiguration eines SOAP-Webservices im Document-Modus

```
<Service type="Soap">
    <SoapVersion>
        <SOAP11/>
        <SOAP12/>
    </SoapVersion>
    <Wsdl name="NameOfTheDocumentService" targetPrefix="service" mode="DOCUMENT"/>
    <Namespace prefix="service" namespace="http://example.org/serviceSpace"/>
    <Namespace prefix="exampleSpace" namespace="http://example.org/namespace"
     schemaLocation="http://localhost:8080/X4/httpstarter/ReST/soapService/xsd/
namespace.xsd"/>
    <Namespace prefix="anotherSpace" namespace="http://example.org/different/
namespace"
     schemaLocation="http://localhost:8080/X4/httpstarter/ReST/soapService/xsd/
differentNamespace.xsd"/>
    <!-- DOCUMENT requires unique localnames for the operations -->
    <Operation name="irrelevantName1"
        process="x4db:/1/RestExamples/soap/DirectOutput.wrf">
        <InputHandler type="SOAP-Envelope"/>
        <OutputHandler type="SOAP-Envelope"/>
        <Input>
            <Part name="firstPart" element="exampleSpace:SpacedOperation"/>
        </Input>
        <Output>
            <Part name="firstPart" element="exampleSpace:Operation1"/>
        </Output>
    </Operation>
    <Operation name="irrelevantName2"
        process="x4db:/1/RestExamples/soap/DirectOutput.wrf">
        <InputHandler type="SOAP-Envelope"/>
        <OutputHandler type="SOAP-Envelope"/>
        <Input>
            <Part name="firstPart" element="anotherSpace:SpacedOperation"/>
        </Input>
        <Output>
            <Part name="Output" element="anotherSpace:SpacedOperation"/>
        </Output>
    </Operation>
    <Operation name="irrelevantName3"
        process="x4db:/1/RestExamples/soap/DirectOutput.wrf">
        <InputHandler type="SOAP-Envelope"/>
        <OutputHandler type="SOAP-Envelope"/>
        <Input>
            <Part name="UnSpacedOperation" type="xsd:integer"/>
        </Input>
        <Output>
            <Part name="Value" type="xsd:String"/>
        </Output>
    </Operation>
</Service>
```

5.14.3.3 SOAP-Webservice mit externer WSDL-Definition bereitstellen

Wie Sie mit der X4 BPMS SOAP-Webservices z.B. im RPC-Stil auf Basis einer bestehenden WSDL-Definition bereitstellen, wird im Folgenden beschrieben.

Voraussetzungen

- Grundlegendes Wissen über den Aufbau von SOAP-Webservices
- Die WSDL-Definition muss gültig und über eine für den Konsumenten gültige URL erreichbar sein, z.B. via HTTP.
- Ein Technical Process pro Webservice-Operation, der die Verarbeitung der eingehenden Anfrage übernimmt. Der Technical Process wird dabei über seine `x4db:/-URL` referenziert.
- Ein URL-Part, über den der SOAP-Webservice via HTTP erreichbar sein soll, z.B. über `http://localhost:8080/X4/httpstarter/ReST/`***`MyExternalWSDLWebservice`***. Diesen URL-Part haben Sie bereits typischerweise als Konstante innerhalb der Webservice-Konfiguration definiert, siehe Webservice-Konfiguration grafisch bearbeiten.

Um einen Technical Process als SOAP-Webservice mit externer WSDL-Definition bereitzustellen, gehen Sie wie folgt vor:

1. Unter der gewünschten URL-Ressource (definiert z.B. durch einen konstanten URL-Part) über das Kontextmenü **Add SOAP Service** einen neuen Service-Eintrag vom Typ *SOAP* anlegen und ggf. (Sicherheits-)Eigenschaften anpassen.
2. SOAP-Version festlegen (entweder *SOAP 1.1*, *SOAP 1.2* oder Unterstützung für beide Standards).
3. WSDL-Modus `EXTERNAL` wählen und in **WSDL URL** (Attribut `url="…"`) eine gültige URL zur externen WSDL-Definition angeben.

Sie können auch eine selbsterzeugte WSDL-Definition und eine XSD aus dem Repository bereitstellen. Stellen Sie die WSDL-Definitionsdatei dazu als File-Service via HTTP bereit, siehe File-Webservice bereitstellen.

Beispiel (Auszug aus XML-Speicherformat)

```
<Constant value="externalWsdl">
    <Service type="File" path="x4db:/1/RestExamples/soap/
externalWsdl.xml"/>
</Constant>
<Constant value="xsd">
    <Remainder name="FILE">
        <Service type="File" path="x4db:/1/RestExamples/soap/xsd/
&lt;FILE>"/>
    </Remainder>
</Constant>
```

Anschließend können Sie die File-Webservice-URL in der Definition des SOAP-Webservices z.B. wie folgt referenzieren:

Beispiel (Auszug aus XML-Speicherformat)

```
<Wsdl mode="EXTERNAL" url="http://localhost:8080/X4/httpstarter/Rest/
soapService/externalWsdl"/>
```

4. Auf **Load Configuration from WSDL** klicken, um die WSDL-Definition mit den entsprechenden Webservice-Informationen (SOAP-Version, Service-Präfix, verknüpfte XSD-Schemas und verfügbare Webservice-Operationen) zu laden.
   In der Baumstruktur des *X4 Webservice Configuration Editors* wird für jede in der WSDL-Definition definierten Webservice-Operation ein entsprechend benannter Kindknoten angelegt.
5. Für jede Webservice-Operation im entsprechenden Kindknoten die Konfiguration anpassen:
   - Technical Process, der die Webservice-Anfrage über den X4 Server verarbeitet, über dessen `x4db:/-URL` verknüpfen.
     Im *Webservice Configuration Editor* ist dies per Drag & Drop aus dem *Repository Navigator* möglich.
   - Input Handler und Output Handler für den SOAP-Webservice definieren.
   - Bei Bedarf Sicherheitseinstellungen (*Guest Allowed*, *Token Return* etc.) anpassen.
6. Speichern und Webservice-Konfiguration mit **Reload on Server** anwenden.
   Der SOAP-Webservice ist ab sofort aktiv und kann über dessen Service-URL via HTTP POST aufgerufen werden.

Bei einem Aufruf der Service-URL via HTTP `GET` wird die in **WSDL URL** (Attribut `url`) angegebene URL weitergeleitet (HTTP-Redirect).

Beispiel: Konfiguration des Webservices mit externer WSDL-Definition

```
<Service type="Soap">
    <SoapVersion>
        <SOAP11/>
        <SOAP12/>
    </SoapVersion>
    <Wsdl mode="EXTERNAL" url="http://www.webservicex.com/CurrencyConvertor.asmx?
wsdl"/>
    <Namespace prefix="exampleSpace" namespace="http://example.org/namespace"/>
    <Namespace prefix="anotherSpace" namespace="http://example.org/different/
namespace"/>
    <!-- requires unique qualified names for the operations -->
    <Operation name="exampleSpace:SpacedOperation"
        process="x4db:/1/RestExamples/soap/DirectOutput.wrf">
        <InputHandler type="SOAP-Envelope"/>
        <OutputHandler type="SOAP-Envelope"/>
    </Operation>
    <Operation name="anotherSpace:SpacedOperation"
        process="x4db:/1/RestExamples/soap/DirectOutput.wrf">
        <InputHandler type="SOAP-Envelope"/>
        <OutputHandler type="SOAP-Envelope"/>
    </Operation>
    <Operation name="UnSpacedOperation"
        process="x4db:/1/RestExamples/soap/DirectOutput.wrf">
        <InputHandler type="SOAP-Envelope"/>
        <OutputHandler type="SOAP-Envelope"/>
    </Operation>
</Service>
```

5.14.3.4 SOAP-Webservice ohne WSDL-Definition bereitstellen

Wie Sie mit der X4 BPMS SOAP-Webservices z.B. im RPC-Stil ohne WSDL-Definition via HTTP-POST bereitstellen, wird im Folgenden beschrieben.

Voraussetzungen:

- Grundlegendes Wissen über den Aufbau von SOAP-Webservices
- Ein Technical Process pro Webservice-Operation, der die Verarbeitung der eingehenden Anfrage übernimmt. Der Technical Process wird dabei über seine `x4db:/-URL` referenziert.
- Ein URL-Part, über den der SOAP-Webservice via HTTP erreichbar sein soll, z.B. über `http://localhost:8080/X4/httpstarter/ReST/`***`MyWebservice`***. Diesen URL-Part haben Sie bereits typischerweise als Konstante innerhalb der Webservice-Konfiguration definiert, siehe Webservice-Konfiguration grafisch bearbeiten.

Um einen Technical Process als SOAP-Webservice ohne WSDL-Definition bereitzustellen, gehen Sie wie folgt vor:

1. Unter der gewünschten URL-Ressource (definiert z.B. durch einen konstanten URL-Part) über das Kontextmenü **Add SOAP Service** einen neuen Service-Eintrag vom Typ *SOAP* anlegen.
2. SOAP-Version festlegen (entweder *SOAP 1.1*, *SOAP 1.2* oder Unterstützung für beide Standards).
3. WSDL-Modus `NONE` wählen.
4. Für jede Operation folgende Konfiguration hinterlegen:
   - Als Namen den *Qualified Name* des ersten Elements im Body des SOAP Requests für die Operation hinterlegen.
   - Technical Process, der die Webservice-Anfrage über den X4 Server verarbeitet, über dessen `x4db:/-URL` verknüpfen.
     Im *Webservice Configuration Editor* ist dies per Drag & Drop aus dem Repository Navigator möglich.
   - Input Handler und Output Handler für den SOAP-Webservice definieren.
5. Speichern und Webservice-Konfiguration mit **Reload on Server** anwenden.
   Der SOAP-Webservice ist ab sofort aktiv und kann über dessen Service-URL ausschließlich via HTTP POST aufgerufen werden. Ein Aufruf der Service-URL via HTTP GET gibt keine WSDL-Definition zurück.

Beispiel: Konfiguration des Webservices ohne WSDL-Definition

```
<Service type="Soap">
    <SoapVersion>
        <SOAP11/>
        <SOAP12/>
    </SoapVersion>
    <Wsdl mode="NONE"/>
    <Namespace prefix="exampleSpace" namespace="http://example.org/namespace"/>
    <Namespace prefix="anotherSpace" namespace="http://example.org/different/
namespace"/>
    <!-- requires unique qualified names for the operations -->
    <Operation name="exampleSpace:SpacedOperation"
        process="x4db:/1/RestExamples/soap/DirectOutput.wrf">
        <InputHandler type="SOAP-Envelope"/>
        <OutputHandler type="SOAP-Envelope"/>
    </Operation>
    <Operation name="anotherSpace:SpacedOperation"
        process="x4db:/1/RestExamples/soap/DirectOutput.wrf">
        <InputHandler type="SOAP-Envelope"/>
        <OutputHandler type="SOAP-Envelope"/>
    </Operation>
    <Operation name="UnSpacedOperation"
        process="x4db:/1/RestExamples/soap/DirectOutput.wrf">
        <InputHandler type="SOAP-Envelope"/>
        <OutputHandler type="SOAP-Envelope"/>
    </Operation>
</Service>
```

5.14.3.5 SOAP-Webservices: Input- und Output-XML

Zu jedem SOAP-Webservice lässt sich in der Webservice-Konfiguration ein Input-/Output-Handler (in den Knoten `<InputHandler/>` und `<OutputHandler/>`) definieren. Diese steuern, welche XML-Datenstruktur erwartet bzw. bei der Übergabe an Webservices erzeugt wird.

5.14.3.5.1 Modus SOAP-Envelope: Input-Handler

Wenn als Input-Handler der Modus *`SOAP-Envelope`* gewählt wurde (XML-Speicherformat: `<InputHandler type="SOAP-Envelope"/>`), so wird das komplette XML des SOAP-Requests inklusive Header-Informationen und Inhalt im Body übergeben, beispielsweise:

```
 <?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Header/>
   <soapenv:Body>
      <GetEntry>
         <id>?</id>
      </GetEntry>
   </soapenv:Body>
</soapenv:Envelope>
```

5.14.3.5.2 Modus SOAP-Envelope-Validated: Input-Handler

Wenn als Input-Handler der Modus *`SOAP-Envelope-Validated`* gewählt wurde, so wird das komplette XML-Dokument des SOAP-Requests inklusive Header-Informationen und Inhalt im Body übergeben. Zusätzlich wird jedoch der Input gegen die verknüpfte WSDL-Definition und die darin enthaltenen XML-Schemata validiert.

Bei Validierungsfehlern wird im Ergebnisdokument ein SOAP Fault (z.B. in einem Element `<SOAP-ENV:Fault>`) zurückgegeben, das die entsprechenden Fehlermeldungen enthält.

> ⓘ Dieser Modus funktioniert nur, wenn bei der Service-Konfiguration das Feld **WSDL URL** ausgefüllt und die WSDL mit **Load Configuration from WSDL** verfügbar gemacht wurde. Siehe auch SOAP-Webservice mit externer WSDL-Definition bereitstellen.

5.14.3.5.3 Modus SOAP-Envelope: Output-Handler

Analog zum Input-Handler gilt, wenn als Output-Handler der Modus SOAP-Envelope gewählt wurde (XML-Speicherformat: `<OutputHandler type="SOAP-Envelope"/>`), so ist der Technical Process für die Erstellung des vollständigen SOAP-Dokuments verantwortlich, d.h. der Technical Process muss sich selber um die Erstellung von `SOAP-Envelope`, `SOAP-Header` und `SOAP-Body` kümmern.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
     <SomeHeader />
  </soap:Header>
  <soap:Body>
    <SomeData />
  </soap:Body>
</soap:Envelope/>
```

5.14.3.5.4 Modus SOAP-Content: Input-Handler

Wenn als Input-Handler der Modus `SOAP-Content` gewählt wurde (XML-Speicherformat: `<InputHandler type="SOAP-Envelope"/>`), so wird der erste Knoten des Bodys innerhalb des SOAP-Requests übergeben, beispielsweise:

Beispiel: SOAP-Request

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
     <SomeHeader />
  </soap:Header>
  <soap:Body>
    <GetEntry>
        <id>101</id>
    </GetEntry>
  </soap:Body>
</soap:Envelope/>
```

Beispiel: Prozess-Input

```
<?xml version="1.0" encoding="UTF-8"?>
<GetEntry>
   <id>101</id>
</GetEntry>
```

5.14.3.5.5 Modus SOAP-Content-Validated: Input-Handler

Wenn als Input-Handler der Modus `SOAP-Content-Validated` gewählt wurde, so wird der erste Knoten des Bodys innerhalb des SOAP-Requests übergeben. Zusätzlich wird jedoch der Input gegen die verknüpfte WSDL-Definition und die darin enthaltenen XML-Schemata validiert.

Bei Validierungsfehlern wird im Ergebnisdokument ein SOAP Fault (z.B. in einem Element `<SOAP-ENV:Fault>`) zurückgegeben, das die entsprechenden Fehlermeldungen enthält.

Dieser Modus funktioniert nur, wenn bei der Service-Konfiguration das Feld **WSDL URL** ausgefüllt und die WSDL mit **Load Configuration from WSDL** verfügbar gemacht wurde. Siehe auch SOAP-Webservice mit externer WSDL-Definition bereitstellen.

5.14.3.5.6 Modus SOAP-Content: Output-Handler

Analog zum Input-Handler gilt, wenn als Output-Handler der Modus `SOAP-Content` gewählt wurde (XML-Speicherformat: `<OutputHandler type="SOAP-Content"/>`), so wird das Ergebnis des Technical Process als Inhalt des SOAP-Bodys für die SOAP-Response genommen.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetEntryResponse>
    <Data>
        <customer>
            <customerid>101</customerid>
            <companyname>SoftProject GmbH</companyname>
            <address>Am Erlengraben 3</address>
            <city>Ettlingen</city>
            <postalcode>76275</postalcode>
            <country>Germany</country>
            <phone>+49 (0)7243 | 561 75-0</phone>
        </customer>
    </Data>
</GetEntryResponse>
```

5.14.3.5.7 Transformation des Prozessergebnisses in den Response-Body

Ist das Prozessergebnis `<SomeData/>` und handelt es sich um ein SOAP 1.1 Request, wird folgende Antwort gesendet:

```
<SOAP11:Envelope xmlns:SOAP11="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP11:Body>
    <SomeData/>
  </SOAP11:Body>
</SOAP11:Envelope/>
```

Ist das Prozessergebnis `<SomeData/>` und handelt es sich um ein SOAP 1.2 Request, wird folgende Antwort gesendet:

```
<SOAP12:Envelope xmlns:SOAP12="http://www.w3.org/2003/05/soap-envelope">
  <SOAP12:Body>
    <SomeData/>
  </SOAP12:Body>
</SOAP12:Envelope/>
```

5.14.4 File-Webservice bereitstellen

Zusätzlich zur Möglichkeit, X4-Prozesse als ReSTful-Services via HTTP(S) bereitzustellen, lassen sich auch statische Datei-Ressourcen aus dem Repository als Service via HTTP GET bereitstellen.

Voraussetzungen:

- Eine Datei im Repository, die Sie als File-Service via HTTP zum Abruf bereitstellen möchten. Die Datei wird dabei über seine `x4db:/-URL` referenziert.
- Ein URL-Part, über den die Datei via HTTP(S) erreichbar sein soll, z.B. über `http://localhost:8080/X4/httpstarter/ReST/MyFile`. Diesen URL-Part haben Sie bereits (als Konstante, Variable oder Remainder) innerhalb der Webservice-Konfiguration definiert, siehe Webservice-Konfiguration grafisch bearbeiten.

5.14.4.1 Datei-Ressource via HTTP bereitstellen

1. Unter der gewünschten URL-Ressource (definiert z.B. durch einen konstanten oder variablen URL-Part) über das Kontextmenü **Add File Service** einen neuen Service-Eintrag vom Typ *File* anlegen.
2. Unter **Path** den `x4db:/`-Pfad zur gewünschten Datei im Repository hinterlegen, deren Inhalt als Datei-Ressource via HTTP unverändert bereitgestellt werden soll.

   Im *X4 Webservice Configuration Editor* können Sie eine Datei aus dem Repository auch per Drag & Drop auf das Feld **Path** zuweisen.
3. Bei Bedarf die Sicherheitseinstellungen (*Guest Allowed, Token Return* etc.) anpassen.
4. Speichern und mit **Reload on Server** die Webservice-Konfiguration anwenden.
   Die Änderungen sind sofort aktiv und Ihr File-Webservice wurde bereitgestellt.

5.14.4.2 File-Webservices mit Variablen nutzen

Die Pfadangabe zur Repository-Datei in Attribut `path` können Sie dynamisch aufbauen, indem Sie den File-Service in einem Variablen-Block definieren, etwa für wechselnde Dateinamen oder IDs innerhalb eines variablen URL-Parts. Die Zuordnung erfolgt anhand des Variablennamens, z.B. `<myVar>`.

Beispiel: Definition eines FILE-Services als Konstante (Auszug aus restconfig.xml)

```
<Variable name="myVar">
    <Service type="File" path="x4db:/1/Project/Folder/&lt;myVar&gt;"/>
</Variable>
```

Erläuterungen:

- Element `<Variable/>` definiert in Attribut `name` den Namen einer Variable, z.B. `myVar`.

- In Attribut `path` wird diese Variable an beliebiger Stelle innerhalb der `x4db:/`-URL als Platzhalter referenziert, auch Mehrfachnennungen des Platzhalters sind möglich.
- Der Platzhalter beginnt immer mit einer spitzen öffnenden Klammer < (`<` im XML-Speicherformat), gefolgt vom Variablennamen und einer spitzen schließenden Klammer >.

Beispiel: Variablen-Platzhalter an beliebiger Stelle platziert (Auszug aus restconfig.xml)

```
<Constant value="varied_1">
    <Variable name="VAR">
        <Service type="File" path="x4db:/1/RestExamples/file/varied_1/&lt;VAR>"/>
    </Variable>
</Constant>
<Constant value="varied_2">
    <Variable name="VAR">
        <Service type="File" path="x4db:/1/RestExamples/file/varied_1/&lt;VAR>.png"/>
    </Variable>
</Constant>
```

Beispiel: Kombinierte Variablen (Auszug aus XML-Speicherformat)

```
<Constant value="varied_3">
    <Variable name="TYPE">
        <Variable name="PART">
            <Service type="File"
             path="x4db:/1/RestExamples/file/deep/&lt;TYPE>/&lt;PART>.&lt;TYPE>"/>
        </Variable>
    </Variable>
</Constant>
<Constant value="varied_4">
    <Variable name="VAR">
        <Constant value="image">
            <Service type="File"
             path="x4db:/1/RestExamples/file/varied_1/&lt;VAR>.png"/>
        </Constant>
    </Variable>
</Constant>
```

5.14.4.2.1 File-Webservices mit Remainder nutzen

Wenn Sie einen File-Webservice mit beliebigen URL-Parts definieren möchten, können Sie auf das *Remainder*-Konstrukt zurückgreifen. Als tolerantere Variante einer Variablen bietet dieses die Möglichkeit, beliebig tiefe URL-Parts abzufangen. Analog zu Variablen definieren Sie hier einen Remainder mit einem Namen und referenzieren diesen in Attribut `path`.

Beispiel: Definition eines FILE-Services mit Remainder (Auszug aus XML-Speicherformat)

```
<Remainder name="myRemainderVar">
    <Service type="File"
     path="x4db:/1/Project/Folder/&lt;myRemainderVar>"/>
</Remainder>
```

Erläuterungen:

- Element `<Remainder/>` definiert in Attribut `name` den Namen einer Remainder-Variable, z.B. `myRemainderVar`.
- In Attribut `path` wird diese Variable an beliebiger Stelle innerhalb der `x4db:/`-URL als Platzhalter referenziert. Auch Mehrfachnennungen des Platzhalters sind möglich.
- Der Platzhalter beginnt immer mit einer spitzen öffnenden Klammer < (bzw. `<` im XML-Speicherformat), gefolgt vom Variablennamen und einer spitzen schließenden Klammer >.

Beispiel: Weitere Beispiele für Remainder (Auszug aus XML-Speicherformat)

```
<Constant value="deep_1">
    <Remainder name="VAR">
        <Service type="File"
         path="x4db:/1/RestExamples/file/deep/&lt;VAR>"/>
    </Remainder>
</Constant>
<Constant value="deep_2">
    <Remainder name="VAR">
        <Service type="File"
         path="x4db:/1/RestExamples/file/deep/&lt;VAR>.png"/>
    </Remainder>
</Constant>
```

5.14.4.2.2 Caching von bereitgestellten Datei-Ressourcen

Jede Datei-Ressource, die über einen File-Webservice per HTTP GET bereitgestellt wird, ist 10 Stunden gültig. Der Client (z.B. ein Web-Browser) ruft die Ressource also erst nach Ablauf dieser Frist erneut auf.

Um für Entwicklungs- und Testzwecke die aktuelle Ressource zu laden, setzen Sie im Client den Cache manuell zurück oder deaktivieren diesen.

5.15 Webservices über Open API bereitstellen

Mit der zentralen Webservice-Schnittstelle lassen sich Technical Processes als Webservices über Open API bereitstellen.

Basis-URL und Funktionsweise

Die Basis-URL der zentralen Webservice-Schnittstelle lautet `http://<server>/X4/httpstarter/ReST` (z. B. `http://localhost:8080/X4/httpstarter/ReST`).

Konfiguration

Webservices werden im X4 Designer über den grafischen *Webservice Configuration Editor* konfiguriert. Weitere Informationen hierzu finden Sie unter Webservice-Konfiguration grafisch bearbeiten.

5.15.1 Webservice-Konfiguration grafisch bearbeiten

Die Webservice-Konfiguration lässt sich im *X4 Designer* über den grafischen *Webservice Configuration Editor* bearbeiten und im *X4 Server* aktualisieren. Technical Processes können so mit wenigen Mausklicks als OpenAPI-basierte Webservices bereitgestellt werden.

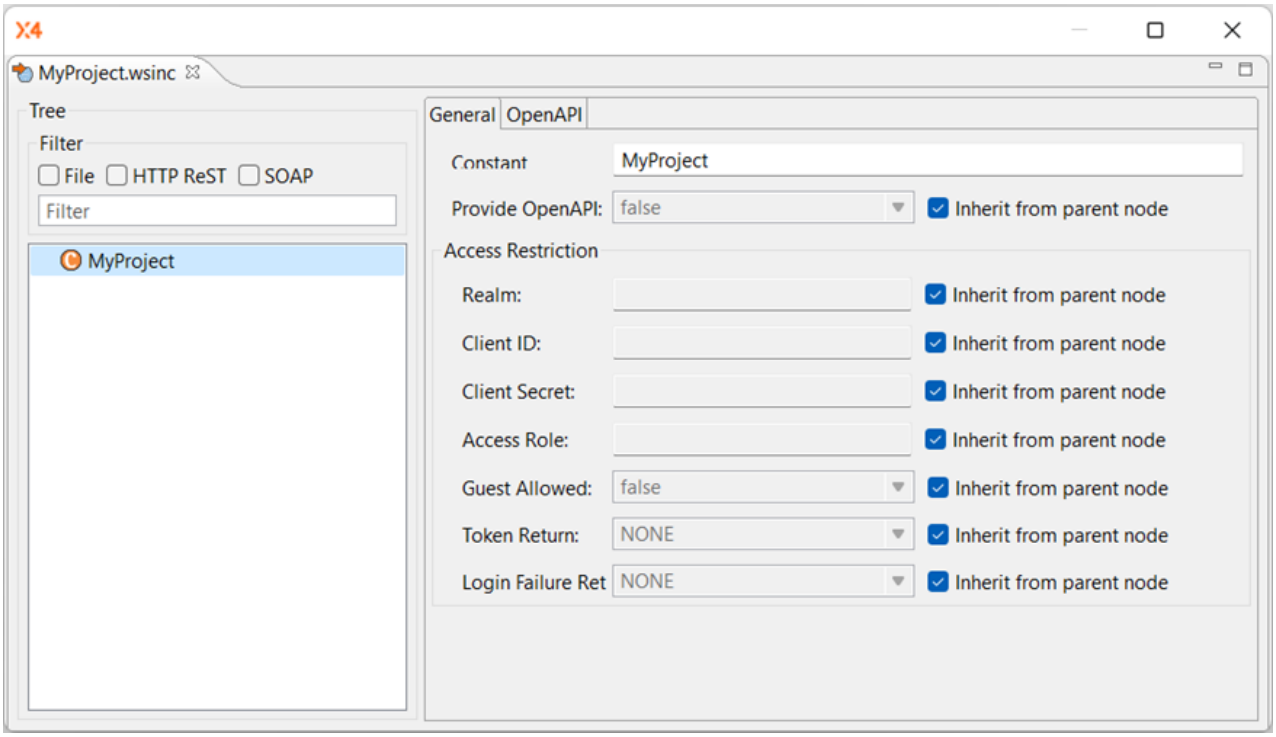
Alle Webservices, die auf diesem Weg bereitgestellt werden, besitzen eine einheitliche Basis-URL. Die Basis-URL ist standardmäßig http://localhost:8080/X4/httpstarter/ReST.

5.15.1.1 Webservice Configuration Editor aufrufen

Um den *Webservice Configuration Editor* aufzurufen, doppelklicken Sie in der Sicht **Projects** auf die `.wsinc`-Datei.

5.15.1.2 Oberfläche des Webservice Configuration Editors

Der *Webservice Configuration Editor* ist in folgende Bereiche unterteilt:



5.15.1.2.1 Tree: URL-Baumstruktur

Im Bereich **Tree** wird die hierarchische Webservice-URL-Struktur bzw. werden die URL-Parts in einer auf- und zuklappbaren Baumstruktur angezeigt. Diese Baumstruktur können Sie per Rechtsklick über das Kontextmenü bearbeiten. Zudem können Sie Elemente innerhalb des Baums per Drag&Drop verschieben.

Über das Feld **Filter** können Sie die angezeigte Baumstruktur anhand des Knotennamens filtern, sodass Strukturen, die das Suchwort nicht enthalten, ausgeblendet werden. Über

oder durch Leeren des Feldes können Sie den Filter zurücksetzen. Zudem kann hier mit `File`, `HTTP ReST` und `SOAP` auch anhand des Service-Typs gefiltert werden.

Baumelemente

Constant	Konstanter URL-Part Sämtliche Eigenschaften der ausgewählten Konstante können Sie im Bereich **Scope** definieren oder Sicherheitseigenschaften vom übergeordneten Element erben lassen.
Variable	Variabler URL-Part Sämtliche Eigenschaften zur ausgewählten Variable können Sie im Bereich **Scope** definieren oder Sicherheitseigenschaften vom übergeordneten Element erben lassen.
Remainder	Komplett beliebige URL-Part-Definition Sämtliche Eigenschaften zum ausgewählten Remainder können Sie im Bereich **Scope** definieren oder Sicherheitseigenschaften vom übergeordneten Element erben lassen.

5.15.1.3 Konstanten, Variablen oder Remainder erstellen und verwalten

Markieren Sie in der Baumstruktur den gewünschten Knoten (entspricht einem URL-Part des Webservices), klicken Sie mit der rechten Maustaste darauf, und wählen Sie im Kontextmenü eine Aktion:

Add Constant	Konstante als Kindknoten anlegen, um einen konstanten URL-Part für Ihren Webservice zu definieren
Add Variable	Variable als Kindknoten anlegen, um mit Platzhaltern variable Bestandteile innerhalb der URL Ihres Webservices zu definieren
Add Remainder	Remainder als Kindknoten anlegen, um alle restlichen URL-Parts abzufangen
Copy Service URL to Clipboard	Zugehörige Webservice-Ressourcen-URL in die Zwischenablage kopieren

Rename	Markierten Knoten umbenennen
Delete	Markierten Knoten löschen

Bei den Knotennamen wird zwischen Groß- und Kleinschreibung unterschieden!

5.15.1.4 Sicherheitseigenschaften des Knotens definieren

Für den ausgewählten Knoten können Sie in **Access Restrictions** bei Bedarf (Sicherheits-)Eigenschaften definieren.

Mit **Inherit from parent node** wird der Sicherheitsbereich vom übergeordneten Element übernommen. Deaktivieren Sie diese Eigenschaft, um einen benutzerdefinierten Wert zu setzen.

Realm	Sicherheitsbereich für den aktuellen URL-Part
Client ID	Keycloak-Client
Client Secret	Keycloak-Client-Anmeldedaten
Access Role	Keycloak-Role Die Role muss in Keycloak dem Benutzer zugewiesen werden, um Zugang zum Webservice zu erhalten.
Guest Allowed	Wenn ein Sicherheitsbereich in Eigenschaft `Realm` definiert wurde, kann mit *`true`* ein Gastzugriff erlaubt werden
Token Return	Zurückgegebenes Log-in-Token auswählen **Mögliche Werte:** • *`NONE`* (Standard) • *`CUSTOM_HEADER`*
Login Failure Return	In Kombination mit dem Log-in-Modul für *X4 Web Apps* können Sie mit **Login Failure Return** steuern, ob nach einem gescheiterten Anmeldeversuch auch der Grund zurückgegeben werden soll. **Mögliche Werte:** • *`NONE`* (Standard): keine Information über den Grund des Scheiterns zurückmelden (wird im Server-Log protokolliert) • *`HEADER`* : den technischen oder fachlichen Grund im HTTP-Header `X-X4-Login-Failed` ausgeben.

5.15.2 ReSTful Webservices bereitstellen

Für jeden URL-Part Ihres ReSTful-Webservices können Sie pro HTTP-Methode jeweils einen verarbeitenden Technical Process zuordnen, z. B. für die *GET*-Methode einen Eintrag `GET`.

Verfügbare HTTP-Methoden

Innerhalb einer Operations-Zuordnung können folgende HTTP-Methoden jeweils einmal definiert werden:

- `GET`
- `PUT`
- `POST`
- `DELETE`
- sowie beliebige weitere Methoden, die einen gültigen XML-Namen besitzen

5.15.3 Technical Process als OpenAPI-ReST-Service bereitstellen

Voraussetzung:
Sie verfügen über ein ESB Project mit mindestens einem Technical Process.

Um im Webservice Configuration Editor einen Technical Process als Operation eines OpenAPI-ReST-Webservice bereitzustellen, gehen Sie wie folgt vor:

1. Geben Sie die gewünschten Daten auf der Registerkarte **OpenAPI** ein.

   ⚠ **Name**, **Title** und **Version** sind obligatorische Felder, in die Sie einen Wert eingeben müssen. Wenn das API eine Methodenbeschreibung anzeigen soll, geben Sie in das Feld **Description** eine Beschreibung ein.

2. Legen Sie in der Baumstruktur bei Bedarf einen neuen Knoten an (entspricht einem URL-Part des Webservices). Weitere Informationen hierzu finden Sie unter Webservice-Konfiguration grafisch bearbeiten.
3. Legen Sie für den gewünschten Knoten (vom Typ `Constant`, `Variable` oder `Remainder`) über **Add HTTP ReST Service** einen neuen Service vom Typ *HTTP ReST* an. Passen Sie ggf. die (Sicherheits-)Eigenschaften an.

   ⚠ Bevor Sie einen HTTP-ReST-Service anlegen, müssen Sie auf der Registerkarte **General** den Wert **Provide OpenAPI** auf **true** setzen und das Kontrollkästchen **Inherit from parent node** deaktivieren. Damit geben Sie an, dass Sie OpenAPI im HTTP-ReST-Service verwenden.

4. Legen Sie für den HTTP-ReST-Service über das Kontextmenü **Add HTTP ReST Operation** eine neue Operation an, oder wählen Sie in der Tabelle **Operations** einen bestehenden Eintrag aus, der überschrieben werden soll.

   ⓘ Sie können jeweils einmal pro Service eine Operation `GET`, `POST`, `PUT`, `DELETE` vergeben oder einen frei wählbaren Operationsnamen eingeben. Bei den Operationsnamen wird zwischen Groß-/Kleinschreibung unterschieden.

5. Ziehen Sie einen Prozess aus der Ansicht **Projects** per Drag&Drop auf die gewünschte Operation, um den Prozesspfad zuzuordnen.
6. Legen Sie über die Dropdown-Liste **Output Mode** den Output-Modus `HTTP-Content` fest, um die Prozess-Response anzuzeigen.
7. Wenn der ReST-Service Query-Parameter erfordert, können Sie diese Parameter anlegen. Dazu klicken Sie in der Registerkarte **OpenAPI** im Gruppenfeld **Query Parameter** auf die Schaltfläche **Add**. Beim Hinzufügen neuer Parameter müssen Sie einen Namen für den Parameter und weitere Optionen wie **Type**, **Array** und **Required** angeben.
8. Über die Registerkarten **Request** und **Response** können Sie einen benutzerdefinierten Request und eine benutzerdefinierte Response für die jeweilige Methode anlegen.

5.15.3.1 Mit einem Datenmodell arbeiten

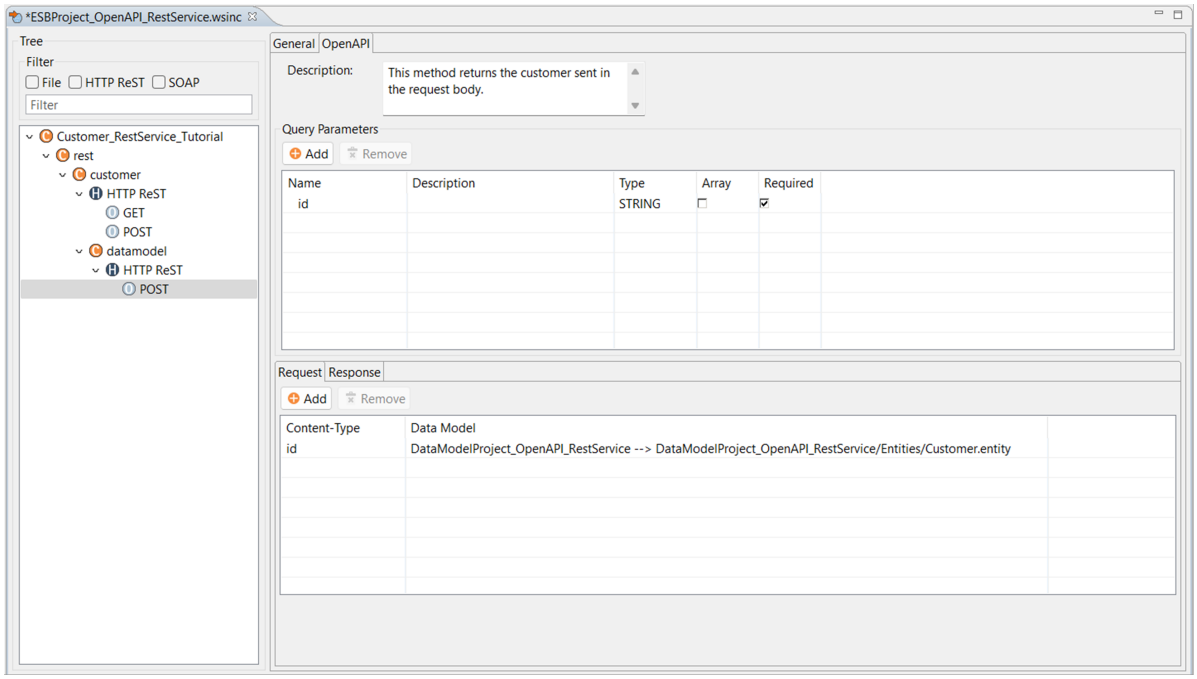
Sie können Ihren ReST-Service so konfigurieren, dass er Werte in Tabellenform aus einem Datenmodell übernimmt und Werte in Tabellenform ausgibt. Dazu müssen Sie zunächst im X4 Designer ein Data Model Project mit mindestens einer Entity anlegen. Jede Entity muss mindestens ein Feld enthalten. Weitere Informationen hierzu finden Sie unter Mit Data Model Projects arbeiten.

> ⓘ Die Datenübernahme aus einem Data Model Project ist für die Methoden `POST`, `PUT` und `DELETE` möglich.

5.15.3.2 Request mit Data Model

Um festzulegen, dass die Eingabedaten für einen Technical Process im Tabellenformat vorliegen sollen, führen Sie die folgenden Schritte auf der Registerkarte **OpenAPI** aus:

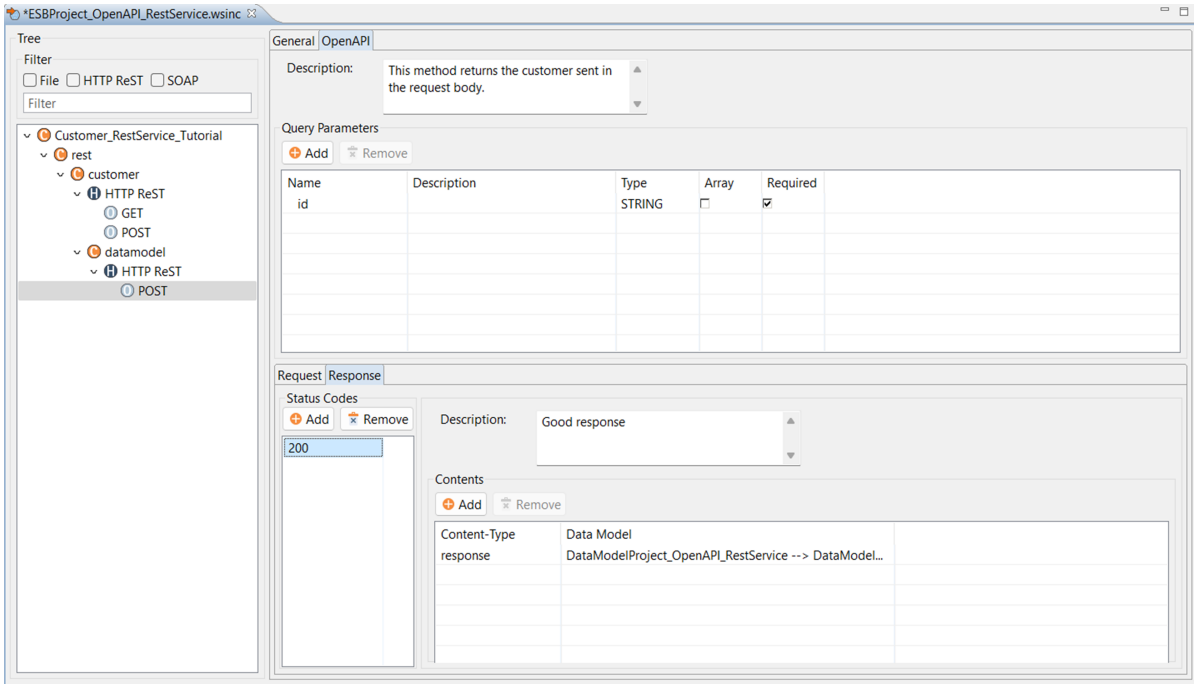
1. Klicken Sie auf der Registerkarte **Request** auf die Schaltfläche **Add**, und geben Sie im Feld **Content-Type** einen Namen ein. Klicken Sie dann in das Feld **Data Model**, und wählen Sie die Schaltfläche mit den drei Punkten.
2. Wählen Sie im Dialogfenster **Select Entities** in der Dropdown-Liste **Data Model** das gewünschte Data Model Project aus, und aktivieren Sie das Kontrollkästchen für die Tabelle, deren Werte Sie übernehmen möchten. Die Tabelle entspricht der Entity, die Sie im Data Model Project angelegt haben.



5.15.3.3 Response mit Data Model

Wenn auch die Ausgabedaten eines Prozesses das Format einer Tabelle haben soll, führen Sie die folgenden Schritte auf der Registerkarte **OpenAPI** aus:

1. Klicken Sie im Bereich **Status Codes** auf die Schaltfläche **Add**, und geben Sie den jeweiligen HTTP-Statuscode an.
2. Geben Sie eine Beschreibung in das Feld **Description** ein.
3. Klicken Sie im Bereich **Contents** auf **Add**, um Inhalte zur Response hinzuzufügen.
4. Klicken Sie im Feld **Data Model** auf die Schaltfläche mit den drei Punkten.
5. Wählen Sie im Dialogfenster **Select Entities** das gewünschte Data Model Project aus, aktivieren Sie das Kontrollkästchen für die gewünschte Tabelle, und klicken Sie auf **OK**.



5.15.3.4 Authentifizierung festlegen

Um eine Methode als privat festzulegen, markieren Sie die gewünschte Methode im Webservice Configuration Editor, und gehen Sie wie folgt vor:

1. Deaktivieren Sie für das Feld **Realm** das Kontrollkästchen **Inherit from parent node**.
2. Geben Sie den gewünschten Wert in das Feld **Realm** ein, z. B. `X4Realm`.

5.15.3.5 ReST-Service testen

Um das konfigurierte OpenAPI im Browser zu öffnen, gehen Sie wie folgt vor:

1. Rufen Sie die URL http://localhost:8080/X4/httpstarter/ReST/[ESB_CONSTANT_VALUE]/openapi.html auf. Ersetzen Sie **[ESB_CONSTANT_VALUE]** durch den Namen Ihres Projekts. Alternativ können Sie mit der rechten Maustaste auf die Methodenoperation klicken und im Kontextmenü **Copy OpenAPI URL to Clipboard** wählen. Fügen Sie die kopierte URL dann im Browser ein.
2. Erweitern Sie unter **Service** den Eintrag für eine Methode, und klicken Sie auf **Try it out**.

> ⚠ Wenn Ihr ReST-Service Parameter enthält, müssen Sie diese unter **Parameter** eingeben, bevor Sie auf **Execute** klicken.

> ⚠ Wenn Ihr ReST-Service ein Data Model Project enthält, befinden sich die Daten im Request-Body. Sie müssen den benötigten Wert eingeben, bevor Sie auf **Execute** klicken.

3. Klicken Sie auf **Execute**.

   Das Ergebnis der Operation wird unter **Responses** angezeigt.

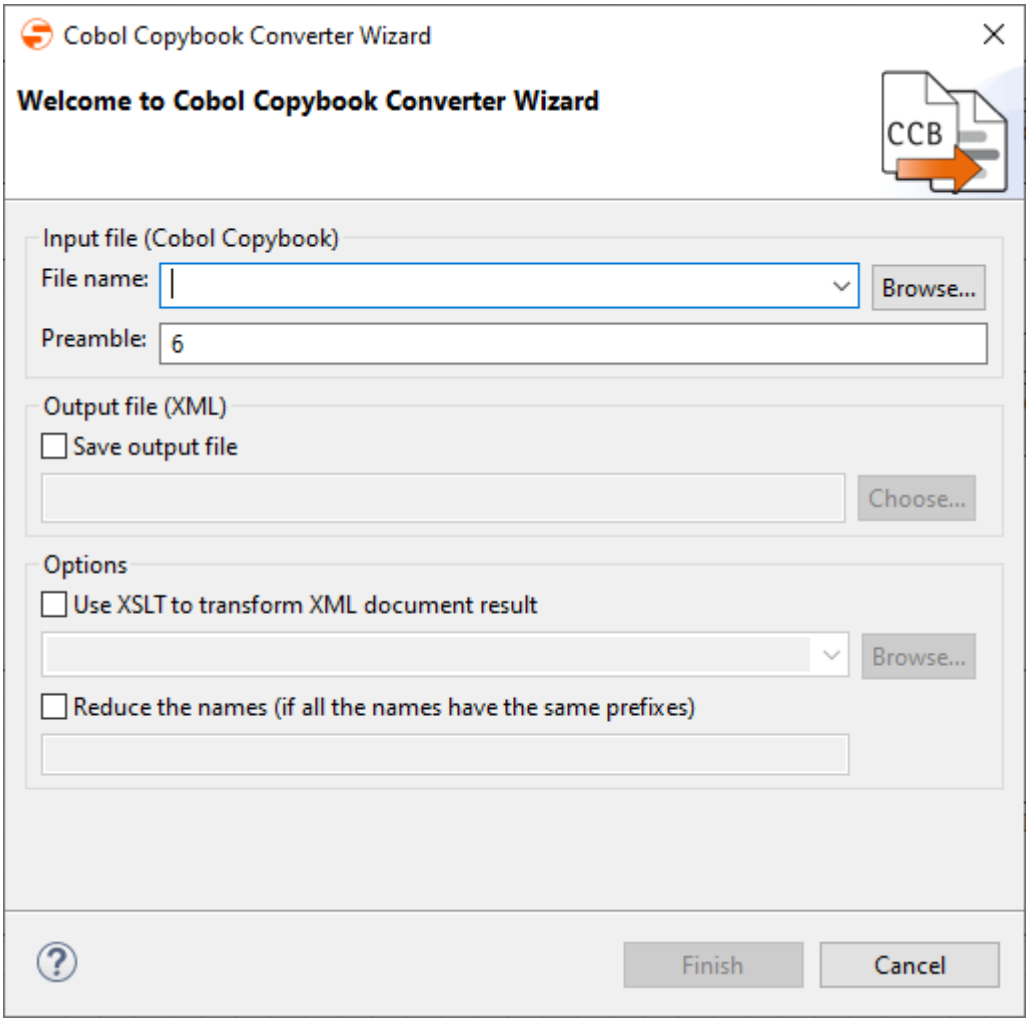
5.15.3.6 ReST-Service mit Authentifizierung testen

Wenn für einen ReST-Service eine Authentifizierung festgelegt wurde, erkennen Sie das daran, dass die Methode auf der rechten Seite mit einem Schloss-Symbol versehen ist. Wenn Sie die Methode öffnen, auf die Schaltfläche **Try it out** klicken, ggf. einen Parameter eingeben und auf **Execute** klicken, wird ein Dialogfenster angezeigt, in dem Sie Ihren Benutzernamen und Ihr Passwort eingeben müssen. Nach Eingabe der Zugangsdaten wird die Methode ausgeführt, und die Response wird angezeigt.

5.16 COBOL Copybook Converter Wizard

Mit dem *COBOL Copybook Converter Wizard* können Sie eine Textdatei im *COBOL Copybook*-Format in ein XML-Dokument konvertieren und im selben Schritt eine XSLT-Transformation der XML-Daten durchführen. Wenn alle Namen denselben Präfix besitzen, können Sie außerdem im Ergebnis-XML-Dokument alle Namen mit demselben Präfix reduzieren.

> ⓘ Um den *COBOL Copybook Converter Wizard* aufzurufen, im Menü **Project > Cobol Copybook Converter...** wählen.



5.16.1 Voraussetzungen

Damit der Wizard korrekt ausgeführt werden kann, muss der Ordner `<X4>\X4DB\999\CCB2XML` auf dem *X4 Server* vollständig vorhanden sein.

5.16.2 Funktionen im COBOL Copybook Wizard

File name	Pfad und Dateiname der *COBOL Copybook*-Datei im Repository
Preamble	Präambel der *COBOL Copybook*-Datei (Standard ist *6*)
Display result of conversion	Vorschau des erzeugten XML-Dokuments im internen Browser anzeigen
Save output file	Ergebnis in einer Datei speichern, deren Repository-Pfad Sie über **Browse** wählen können
Use XSLT to transform ...	Eine XSL-Transformation vor dem Speichern auf das erzeugte XML-Dokument anwenden
Reduce the names ...	Alle Namen mit demselben Präfix reduzieren

5.17 SessionID einer Web App auslesen

Wenn ein Benutzer auf eine Web App zugreift, können die Benutzerdaten in einem Technical Process mithilfe des **User Identity Adapters** ausgelesen werden.

Die ausgelesenen Benutzerdaten enthalten unter anderem die `SessionID`, die der Benutzer für diese Sitzung verwendet. Eine `SessionID` ist ein eindeutiger Sitzungsbezeichner. Mit der `SessionID` kann der Benutzer und die Sitzung eindeutig einander zugeordnet werden. So kann Daten in die Sitzung abgelegt oder auf die Daten der Sitzung zugegriffen werden. Die SessionID einer Web App kann aus dem sogenannten **JSON Web Token** ausgelesen werden.

Die `SessionID` sind in der Output-XML des **User Identity Adapters** enthalten.

Weitere Informationen finden Sie im Abschnitt User Identity Adapter.

5.18 Gruppen eines Benutzers auslesen

Wenn ein Benutzer auf eine Web App zugreift, können die Benutzerdaten in einem Technical Process mithilfe des **User Identity Adapters** ausgelesen werden.

Die ausgelesenen Benutzerdaten enthalten unter anderem die Gruppen, die dem Benutzer zugewiesen sind. Die Gruppen sind in der Output-XML des **User Identity Adapters** enthalten.

Weitere Informationen finden Sie im Abschnitt User Identity Adapter.

5.19 Claims eines Benutzers auslesen

Wenn ein Benutzer auf eine Web App zugreift, können die Benutzerdaten in einem Technical Process mithilfe des **User Identity Adapters** ausgelesen werden.

Die ausgelesenen Benutzerdaten enthalten unter anderem die Claims eines Benutzers. Die Claims sind in der Output-XML des **User Identity Adapters** enthalten.

Weitere Informationen finden Sie im Abschnitt User Identity Adapter.

6 Prozesse debuggen und ausführen

Erfahren Sie hier, wie Sie Technical Processes und Business Processes auf Fehler überprüfen (debuggen) und auf dem X4 Server ausführen.

6.1 Technical Process debuggen

Während der Prozess-Entwicklung können Sie direkt im Process Editor Ihre Technical Processes (und unter bestimmten Voraussetzung auch Business Processes) auf syntaktische Fehler der einzelnen Prozessbausteine überprüfen.

Voraussetzungen:

- Mit allen Prozessbausteinen ist, falls erforderlich, eine Repository-Datei verknüpft.
- Bei allen Prozessbausteinen sind sämtliche Pflichtparameter und Eigenschaften vollständig parametriert.
- Für Business Processes: Der Business Process wurde nicht im Free-flow-Modus angele gt.

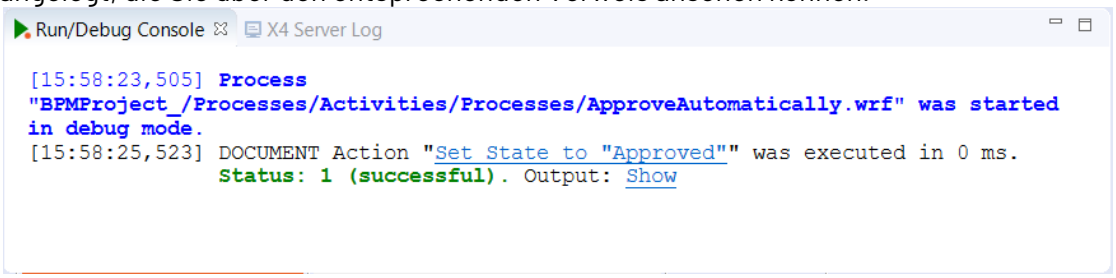
1. Falls gewünscht, Debug-Haltepunkte (Breakpoints) setzen, um bei kritischen Prozessschritten das Debugging automatisch anzuhalten:
   - Im Prozessdiagramm den gewünschten Prozessbaustein anklicken.
   - In der Symbolleiste **Toggle Breakpoint** klicken.
2. **Start Debugging (F4)** klicken, um den Debug-Modus zu starten. Um eine parallele Debug-Session zu starten, erneut **Start Debugging** klicken.

   Sie können für alle Technical Process eine parallele Debug-Session starten.
   In der Sicht **Debug Sessions** sehen Sie alle aktiven Debug-Sessions. Doppelklicken Sie in der Sicht **Debug Sessions** auf den gewünschten Technical Process, um in die Debug-Session zu wechseln.
   Aktive Debug-Sessions sind grau markiert. Mit dem Server verbundene, neue Debug-Sessions sind orange markiert.

3. Nächsten Schritt ausführen.
   a. **Step Into (F5)** klicken, um den nächsten Einzelschritt auszuführen und anzuzeigen. Der Debugger springt in Subprozesse. Jeder Einzelschritt im Subprozess muss ausgeführt werden.
   b. **Step Over (F6)** klicken, um den nächsten Einzelschritt auszuführen und anzuzeigen. Der Debugger überspringt Subprozesse, stopp jedoch bei Erreichen eines Haltepunkts. Das Ergebnis des Subprozesses wird als Output angezeigt.
   c. **Step Return (F7)** klicken, um alle Schritte in einem Subprozess bis zum Ende oder Haltepunkt auszuführen und anzuzeigen. Der Debugger springt anschließend in den Hauptprozess.
   d. **Resume (F8)** klicken, um alle Schritte bis zum Ende oder Haltepunkt auszuführen und anzuzeigen.

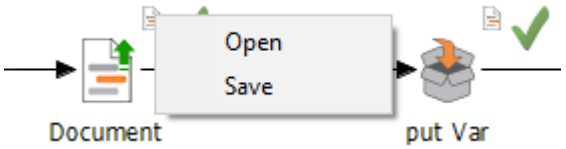
e. **Resume without debugging (F10)** klicken, um das Debugging zu beenden und alle Schritte auszuführen. Haltepunkte werden übersprungen.

f. In einem Technical Process ein Element rechtsklicken. Im Kontextmenü **Run to element** klicken, um alle Schritte bis zum Element oder Haltepunkt auszuführen und anzuzeigen. Zu jedem ausgeführten Prozessschritt wird ein Eintrag in der Sicht **Run/Debug Console** angezeigt. Fehlermeldungen werden rot hervorgehoben, erfolgreiche Statusmeldungen sind grün. Außerdem wird jeweils eine temporäre Ausgabedatei angelegt, die Sie über den entsprechenden Verweis ansehen können.



Zusätzlich wird die jeweilige Ausgabedatei der entsprechenden Prozesskomponente im Prozessmodell angehängt. Über das Kontextmenü lässt sich die Ausgabedatei zudem abspeichern.

> Enthält das Ergebnis Inhalte, die größer als 1 MB sind, wird eine Meldung ausgegeben und das Ergebnis lässt sich über den Dialog herunterladen.



> Die Sicht **Output** und die Sicht **Quick Watch** bieten zusätzliche Möglichkeit beim Debuggen:
>
> - **Output**: Die Ergebnisse jedes einzelnen Schrittes lassen sich während des Debuggens einsehen. Große Dateien können über eine Download-Schaltfläche heruntergeladen werden.
> - **Quick Watch**: Platzhalter, dynamischen Parameter und Kontextvariablen lassen sich während des Debuggens betrachten. Große Dateien können über eine Download-Schaltfläche heruntergeladen werden.
>
> Beide Sichten lassen sich über das Menü **Debug > Output** bzw. **Quick Watch** oder über die jeweiligen Icons (Output) und (Quick Watch) in der Symbolleiste aufrufen.

4. Mit **Terminate (STRG+F2)** können Sie den Debug-Modus bei Bedarf manuell beenden. Wenn ein technischer Fehler auftritt, dann wird der Debug-Modus automatisch abgebrochen.

Weitere Informationen finden Sie in diesen Abschnitten:

- Debug-Session ohne Debugging ausführen
- Debug-Session beenden
- Debugger mit Server verbinden

6.1.1 Debug-Session ohne Debugging ausführen

Sie können eine oder mehrere Debug-Sessions ohne Debugging ausführen und beenden.

1. Checkbox aktivieren, um den Prozess auszuwählen.
2. **Resume without debugging** klicken, um alle Schritte auszuführen und das Debugging zu beenden. Haltepunkte werden übersprungen.

Weitere Informationen finden Sie im Abschnitt

- Debug-Session beenden
- Debugger mit Server verbinden

6.1.2 Debug-Session beenden

Sie können eine oder mehrere Debug-Sessions ohne Debugging beenden.

1. Checkbox aktivieren, um den Prozess auszuwählen.
2. **Terminate** klicken, um das Debugging zu beenden.

Weitere Informationen finden Sie im Abschnitt

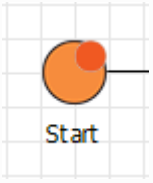
- Debug-Session ohne Debugging ausführen
- Debugger mit Server verbinden

6.1.3 Debugger mit Server verbinden

Der Server kann nur mit einem X4 Designer zur selben Zeit verbunden sein.
Sobald der Server Daten an den Debugger sendet und ein Haltepunkt erreicht wird, blinkt der X4 Designer.

Um Prozesse mit Anfragen oder Daten zu prüfen, die vom Server erhalten werden, muss der Debugger mit dem Server verbunden werden. Zum Debuggen werden Anfragen und Daten des Servers verwendet, eine Puffer-Datei wird nicht benötigt.

Um die vom Server gesendeten Daten zu sehen, setzen Sie einen Haltepunkt an dem Start-Baustein.



1. **Attach Debug** klicken, um den Debugger mit der Server-Instanz zu verbinden.
2. **Detach Debug** klicken, um den Debugger von der Server-Instanz zu lösen.

Weitere Informationen finden Sie im Abschnitt

- Debug-Session ohne Debugging ausführen
- Debug-Session beenden

6.2 Prozess ausführen

Sie können Technical Processes oder Business Processes direkt aus dem X4 Designer heraus starten. Dabei wird der Technical Process auf dem X4 Server ausgeführt und es werden keine Debug-Informationen im Server-Protokoll angezeigt.
Um Technical Processes oder Business Processes auszuführen, gibt es mehrere Möglichkeiten:

Geöffneten Prozess starten	In der Symbolleiste auf **Run Without Debugging** klicken, um den Process zu starten.
Im Repository Navigator	Den Process rechtsklicken, im Kontextmenü **Run Without Debugging** wählen.
Als Subprozess	Dieser wird automatisch vom übergeordneten Technical Process aufgerufen.
Zeitgesteuert ausführen	Mithilfe des Schedulers Processes zu festgelegten Zeiten starten

6.3 Statuswerte für Prozesse und Prozess-Schritte

Während des Debugging und beim Ausführen eines Prozesses liefert der X4 Server Status-Werte für jeden Prozess-Schritt und für ausgeführte Prozesse. Als Statuswerte werden beliebige ganze Zahlen (positiv, negativ, 0) angezeigt.

Hinweis: Prüfen Sie die Statuswerte des Adapters, den Sie im Prozess verwenden, um sicherzustellen, dass die Statuswerte des Adapters nicht mit weiteren Statuswerten im Prozess kollidieren.

Empfehlung: Verwenden Sie für benutzerdefinierte Statuswerte Zahlen größer als 1000, um mögliche Überschneidungen mit anderen Statuswerten zu vermeiden.

Zahl	Label	Bedeutung
`-999`		Fehlerstatus der Engine Auftreten: • Bei einem Engine-Problem erscheint die Meldung: `No branch found to continue` • Prozess kann nicht gestartet werden, weil der Prozess defekt ist
`-998`		Die Ausführung des Prozesses wurde aus Lizenzgründen nicht ausgeführt. Mögliche Gründe: • keine Lizenz • eine beschädigte Lizenz • eine abgelaufene Lizenz • eine Lizenz, die nicht zur Serverversion passt
`-1`	`error`	Fehler bei der Ausführung des Prozessbausteins (Details werden im Server-Log angezeigt) Als PID (Prozess-Instanz-ID) wird angezeigt: Der Prozess ist "Out of Service"
`0`	`successful without result`	Der Prozessbaustein wurde ausgeführt und es gibt kein Ergebnisdokument bzw. das Ergebnis ist Null (leer).
`1`	`successful`	Der Prozessbaustein wurde ohne syntaktische Fehler ausgeführt.
`997`		Der Prozess wurde geplant abrupt angehalten (terminiert) (vergleichbar mit einer nicht abgefangenen Exception in Java). Bewirkt die sofortige Beendigung des gesamten Prozesses.
`998`		Der Prozess wurde (erfolgreich) angehalten und in einem Save Point gespeichert. Bewirkt die sofortige Beendigung des gesamten Prozesses.
`999`		Status-ID von noch nicht ausgeführten Prozessbausteinen
(beliebige andere negative Ganzzahl)		Mittels *ConditionState-Mapping* gesetzter individueller Fehlerstatus; idealerweise 4-stellig; siehe Status mit ConditionState dynamisch setzen
(beliebige andere positive Ganzzahl)		Mittels *ConditionState-Mapping* oder von einem Adapter mit variablen Status-Werten gesetzter individueller Ergebnis-Status; idealerweise 4-stellig; siehe Status mit ConditionState dynamisch setzen

7 Scheduler

Die Scheduler-Oberfläche bietet einen Überblick über sämtliche Prozesse, für die eine zeitgesteuerte Ausführung definiert wurde. Zudem bietet die Oberfläche auch die Möglichkeit, neue zeitgesteuerte Ausführungen zu konfigurieren sowie bereits bestehende Konfigurationen zu bearbeiten oder zu löschen.

Bei der Definition von Schedules wird zwischen folgenden Ausführungsoptionen unterschieden:

- **Serverstart**: Ausführung eines Prozesses beim Serverstart
- **CRON**: Ausführung eines Prozesses nach bestimmten Cron-Mustern

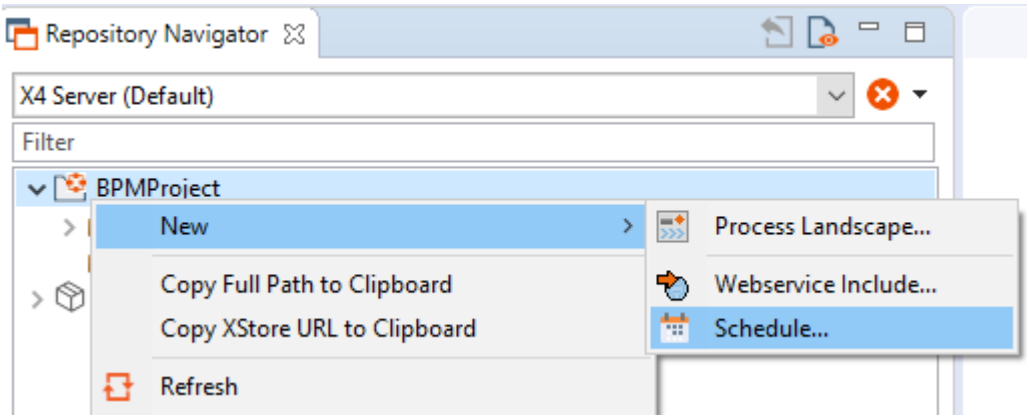
Die wichtigsten Funktionen des *Schedulers* sind:

- Übersicht der bestehenden Schedules
- Erstellen/Bearbeiten von beliebig vielen Ausführungspunkten und -intervallen pro Prozess

7.1 Scheduler erstellen

- Der Scheduler kann nur in **BPM**- und **ESB**-Projekten erstellt werden.
- In einem Projekt kann nur ein Scheduler erstellt werden.

1. In der Repository Navigator Sicht ein Projekt rechtsklicken.
2. Im Kontextmenü **New> Schedule** wählen.



Der Scheduler wird erstellt und geöffnet.

Weitere Informationen finden Sie im Abschnitt

- Prozesse zeitgesteuert ausführen
  - Serverstart
  - Cron

7.2 Prozesse zeitgesteuert ausführen

Bei der zeitgesteuerten Ausführung von X4-Prozessen können folgende Ausführungsoptionen unterschieden werden:

- Ausführung eines Prozesses beim Serverstart
- Ausführung eines Prozesses nach bestimmten CRON-Mustern

7.2.1 Neue zeitgesteuerte Ausführung anlegen

Um eine zeitgesteuerte Ausführung anzulegen, gehen Sie wie folgt vor:

1. Im Scheduler `.schedule` die gewünschte Ausführungsoption klicken.
   a. Cron, um Prozesse nach bestimmten Cron-Mustern auszuführen.
   b. Startup, um Prozesse beim Serverstart auszuführen.
2. Gewünschte Konfiguration vornehmen.

   Für die Konfigurationsmöglichkeiten der verschiedenen Ausführungstypen, siehe:
   - Serverstart
   - Cron
3. **Speichern** klicken, um die Ausführung zu speichern.

Die Dokumentation der ReST-Schnittstelle kann unter http://localhost:8080/X4/X4Api/ im OpenAPI 3.0-Format abgerufen werden.
Um die ReST-Schnittstelle zu nutzen, kann beispielsweise der Swagger Editor verwendet werden.

7.2.2 Serverstart

Neue Konfigurationen für Ausführungen beim Serverstart können über die Schaltfläche **Startup** angelegt werden. Dabei stehen eine Reihe von Optionen und Parameter zur Verfügung. Diese werden im Folgenden näher erläutert.

Optionen

Name	Frei wählbarer Name für die Ausführung

Process	Pfad zum Prozess in der `X4DB` *Beispiel*: `Processes/Activities/Processes/Process.wrf` **Browse...** klicken, um einen Prozess zu wählen. ⚠ • Der Prozess muss sich im selben Projekt wie der Scheduler befinden. • Der Prozess darf kein Unterprozess eines weiteren Prozesses sein.
Active	Aktiviert den Scheduler
Start date	Setzt den Startzeitpunkt des Schedulers im Format `jjjj-mm-tt hh:mm:ss` *Beispiel:* `2021-06-01 13:00:00`
End date	Setzt den Endzeitpunkt des Schedulers im Format `jjjj-mm-tt hh:mm:ss` *Beispiel:* `2022-08-01 23:00:00` ℹ Der Endzeitpunkt muss in der Zukunft liegen.

Parameter

Im Bereich **Parameter** können Sie beliebig viele Parameter zur Ausführkonfiguration definieren.

	Zeile hinzufügen
	Zeile löschen
Key	Name des Parameters (beliebige Zeichenkette, idealerweise `NCName`)
Value	Optionaler Parameterwert

7.2.3 Cron

Neue Konfigurationen für Ausführungen beim Serverstart können über die Schaltfläche **Cron** angelegt werden. Dabei stehen eine Reihe von Optionen und Parameter zur Verfügung. Diese werden im Folgenden näher erläutert.

Optionen

Name	Frei wählbarer Name für die Ausführung
Process	Pfad zum Prozess in der `X4DB` *Beispiel*: `Processes/Activities/Processes/Process.wrf` **Browse...** klicken oder einen Prozess per **Drag & Drop** in das Feld ziehen, um einen Prozess zu wählen. ⚠ • Der Prozess muss sich im selben Projekt wie der Scheduler befinden. • Der Prozess darf kein Unterprozess eines weiteren Prozesses sein.
Active	Aktiviert den Scheduler
Generate Cron Expression	Erzeugt einen Cron-Ausdruck **Mögliche Werte:** • `Minutes` • `Hourly` • `Daily` • `Weekly` • `Monthly` • `Yearly` ⓘ In der Registerkarte Source können Sie den Cron-Ausdruck manuell editieren. Das Generieren eines Cron-Ausdrucks ist nicht möglich, wenn der Cron-Ausdruck manuell editiert wurde.

Parameter

Im Bereich **Parameter** können Sie beliebig viele Parameter zur Ausführkonfiguration definieren.

	Zeile hinzufügen
	Zeile löschen
Key	Name des Parameters (beliebige Zeichenkette, idealerweise `NCName`)
Value	Optionaler Parameterwert

8 Prozessmonitoring

In einer Datei für das Monitoring von Profiling-Daten (`.profiling`) können Sie festlegen, ob Prozesse innerhalb Ihres Projekts zur Laufzeit protokolliert werden sollen. In dieser Datei können Sie dabei genau definieren, welche Prozesse von der Protokollierung erfasst oder ausgeschlossen werden.

Die `.profiling`-Datei muss nur angelegt werden, wenn Sie bestimmte Prozesse von der Protokollierung ausschließen wollen.

So legen Sie eine Datei für das Monitoring von Profiling-Daten an

1. Klicken Sie in der Sicht **Projects** mit der rechten Maustaste auf ein Projekt.
2. Wählen Sie im Kontextmenü **New** > **Monitoring**.
   Die Monitoring-Konfigurationsdatei wird erstellt und geöffnet.

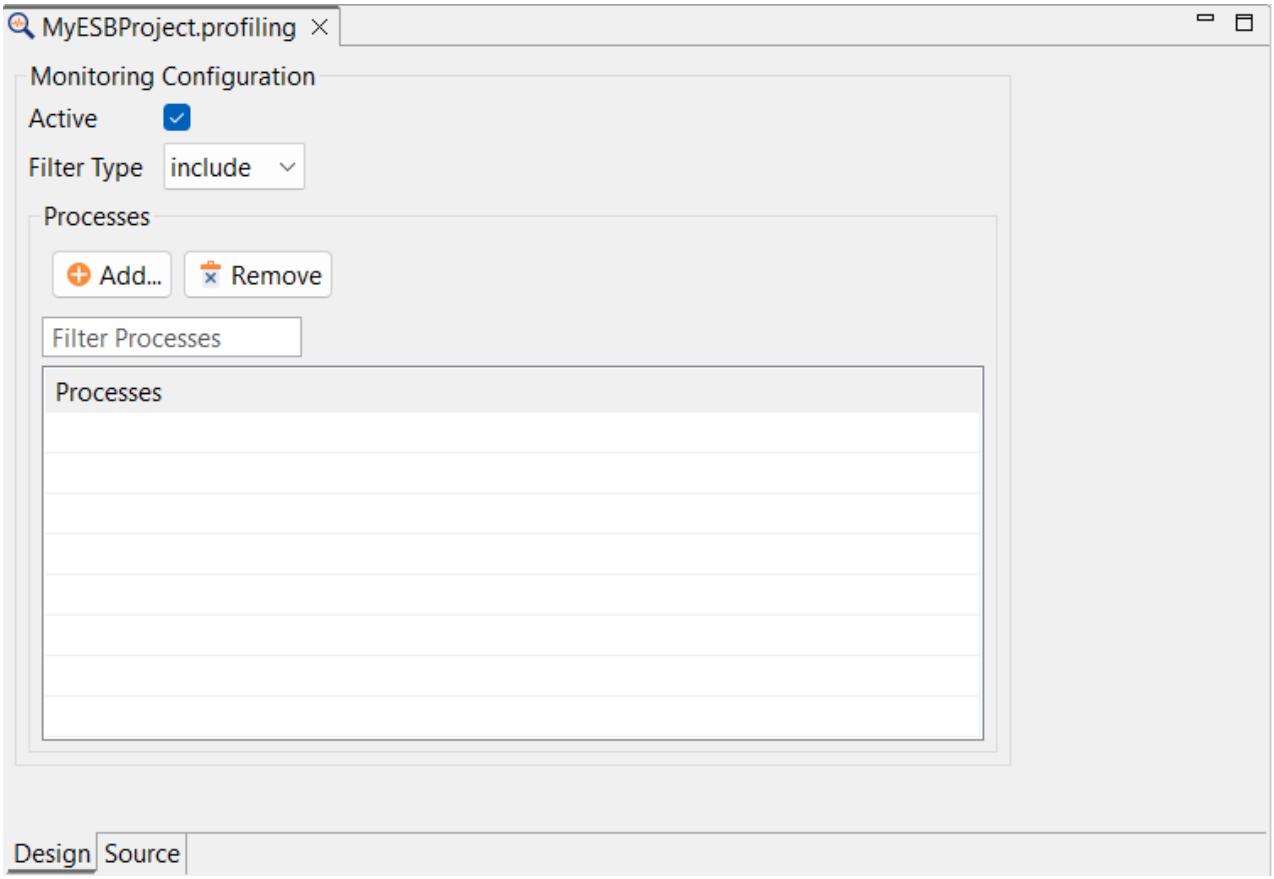
> **Hinweis:**
> In einem Projekt kann nur eine `.profiling`-Datei erstellt werden.

Sie können das Monitoring von Profiling-Daten (`.profiling`-Datei) in der Design-Ansicht und in der Source-Ansicht konfigurieren. Beide Möglichkeiten werden auf den folgenden Seiten beschrieben. Zum Aufrufen der gewünschten Ansicht klicken Sie am unteren Rand einer `.profiling`-Datei auf **Design** oder **Source**.

> **Weitere Informationen:**
> - Monitoring von Profiling-Daten in der Design-Ansicht konfigurieren
> - Monitoring von Profiling-Daten in der Source-Ansicht konfigurieren

8.1 Monitoring von Profiling-Daten in der Design-Ansicht konfigurieren

Wenn Sie das Monitoring in der Design-Ansicht konfigurieren, sehen Sie alle verfügbaren Einstellungen im Fenster **Monitoring Configuration**.



Sie können im Fenster **Monitoring Configuration** die folgenden Einstellungen vornehmen:

Kontrollkästchen **Active**	Mit diesem Kontrollkästchen legen Sie fest, ob die Protokollierung von Prozessen aktiviert oder deaktiviert ist. Standardmäßig ist das Kontrollkästchen aktiviert.
Dropdown-Liste **Filter Type**	Mit dieser Dropdown-Liste geben Sie an, ob die hinterlegten Prozesse von der Protokollierung erfasst oder ausgenommen werden sollen. Verfügbare Optionen: **include:** Die hinterlegten Prozesse werden von der Protokollierung explizit erfasst. **exclude:** Die hinterlegten Prozesse werden von der Protokollierung explizit ausgeschlossen.
Bereich **Processes**	In diesem Bereich können Sie Prozesse hinzufügen (Schaltfläche **Add)** und entfernen (Schaltfläche **Remove**). Außerdem können Sie Filterkriterien für hinzugefügte Prozesse angeben. Für Prozesse, die Sie hier hinzufügen, können Sie angeben, ob diese Prozesse von der Protokollierung erfasst oder ausgenommen werden sollen.

8.2 Monitoring von Profiling-Daten in der Source-Ansicht konfigurieren

Das Wurzelelement der Monitoring-Datei ist `<Profiling>`. Das Element kann einzelne Prozesse des Projekts enthalten, die von einer Protokollierung explizit erfasst oder ausgeschlossen werden sollen.

Das Element `<Profiling>` kann folgendes Attribut besitzen:

Attribut	Beschreibung
`active`	Legt fest, ob die Protokollierung von Prozessen aktiviert oder deaktiviert ist. **Mögliche Werte:** • *`true`*: Die Protokollierung der Prozesse ist aktiviert. (Standard) • *`false`*: Die Protokollierung der Prozesse ist deaktiviert.

Das `<Profiling>`-Element enthält das Kindelement `<Processes>`, das folgendes Attribut enthalten kann:

Attribut	Beschreibung
`filterType`	Legt fest, ob die hinterlegten Prozesse von der Protokollierung erfasst oder ausgenommen werden sollen. **Mögliche Werte:** • *`include`*: Die hinterlegten Prozesse werden von der Protokollierung explizit erfasst. • *`exclude`*: Die hinterlegten Prozesse werden von der Protokollierung explizit ausgeschlossen.

Das Element `<Processes>` kann `<Process>`-Kindelemente enthalten, sofern ein oder mehrere Prozesse explizit von der Protokollierung erfasst oder ausgenommen werden sollen.

Das Element `<Process>` enthält folgendes Attribut:

Attribut	Beschreibung
`path`	*Erforderlich*. Enthält den Pfad zum Technical Process, der erfasst oder ausgeschlossen werden soll. **Mögliche Werte:** Repository-Pfad zu einem Technical Process (z. B. `Processes/Process.wrf`)

8.2.1 Beispiel einer `.profiling`-Datei

Im nachfolgenden Beispiel werden die Technical Processes `Prozess_B.wrf` und `Process_C.wrf` von der Protokollierung ausgeschlossen.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Profiling>
    <Processes filterType="exclude">
        <Process path="Processes/Process_B.wrf"/>
        <Process path="Processes/Process_C.wrf"/>
    </Processes>
</Profiling>
```

9 Mit Adapter-Projekten arbeiten

Innerhalb von Adapter-Projekten lassen sich neue Adapter für die X4 BPMS mit Hilfe der verschiedenen Werkzeuge der X4 BPMS erstellen, siehe Eigene Adapter mit der X4 BPMS entwickeln.

Um diese Funktion der X4 BPMS zu nutzen, wird eine ADK-Lizenz vorausgesetzt.

10 Übersicht der X4 Adapter

10.1 Über X4 Adapter

Die X4 BPMS verfügt über eine umfangreiche Sammlung an Standard-Adaptern zur Integration von Anwendungen und Daten in beliebige Business Processes. Mit ihnen verfolgt die X4 BPMS einen modularen Ansatz. Dabei werden Adapter als Bausteine in Technical Processes integriert und stellen Datenanbindungen her, beispielsweise mit Oracle-Datenbanken, übertragen Daten über verschiedene Protokolle wie HTTP und JMS, oder transformieren und konvertieren verschiedene Daten und Datenformate.

Dieser Lösungsansatz ermöglicht es, heterogene IT-Landschaften in eine Service-orientierte Architektur zu überführen.

Eigenschaften der X4 Adapter:

- Plug-and-Play-Fähigkeit
- Stetig wachsende Anzahl an Adaptern, derzeit mehr als 200
- Adapter für alle gängigen Standardprotokolle, Formate und Systeme
- Einfache Erstellung eigener Adapter durch das X4 Adapter Development Kit (ADK)
- Einfache Nutzung im X4 Designer per Drag&Drop
- Konfigurieren statt programmieren

10.2 Big Data

- Amazon S3 DynamoDB Connector
- Apache Cassandra Connector
- Google Cloud Storage BigTable Connector
- Microsoft Azure Data Lake Gen 1 Storage Connector

10.2.1 Amazon S3 DynamoDB Connector

`Amazon S3 DynamoDB Connector`: Amazon S3 DynamoDB ist ein vollständig verwalteter NoSQL-Datenbankdienst, der eine schnelle und vorhersehbare Leistung bei nahtloser Skalierbarkeit bietet. Der Adapter speichert Daten im Amazon S3 DynamoDB Storage und ermöglicht die Verwendung der X4 BPMS, um Daten in Cloud-Umgebungen zu persistieren.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `CreateTable`: Erstellt eine Tabelle mit Tabellen-Name und Primärschlüssel **Erforderliche Parameter:** `tableName`, `partitionKey`, `partitionType`. Für sortierbare Schlüssel zusätzlich `sortKey`, `sortType`. Für AutoScaling zusätzlich `scalingMinCapacity`, `scalingMaxCapacity`, `scalingTargetValue`, `scalingInCooldown`, `scalingOutCooldown`, `scalableType`, `arnRole` • `UpdateTable`: Aktualisiert vorhandene Tabellen und ändert die Read/Write Capacity Units. **Erforderliche Parameter:** `tableName`, `partitionKey`, `readCapacityUnits`, `writeCapacityUnits` • `ListTable`: Listet alle Tabellen auf. • `EnableAutoScaling`: Aktiviert AutoScaling für eine bestehende Tabelle mit dem entsprechenden `scalableType` (Read/Write). **Erforderliche Parameter:** `tableName`. `scalingMinCapacity`, `scalingMaxCapacity`, `scalingTargetValue`, `scalingInCooldown`. `scalingOutCooldown`, `scalableType`, `arnRole` • `DisableAutoScaling`: Deaktiviert AutoScaling für eine bestehende Tabelle mit dem entsprechenden `scalableType` (Read/Write). **Erforderliche Parameter:** `tableName`, `scalableType` • `PutItem`: Empfängt ein XML-Input mit der Struktur des zu platzierenden Elements **Erforderliche Parameter:** `tableName` • `BatchPutMultipleItems`: Fügt mehrere Elemente in einem einzigen Aufruf ein. **Erforderliche Parameter:** `tableName` **Erforderliche Dateien** Für diese Operation wird eine Input-XML-Datei benötigt, die die Elemente enthält.

Parameter	• `GetItem`: Ruft das spezifische Element ab. **Erforderliche Parameter:** `tableName`, `partitionKey`, `partitionType`, `partitionValue` (muss mit dem Element, das abgerufen werden soll übereinstimmen). Bei sortierbare Schlüssel zusätzlich `sortKey`, `sortType`, `sortKeyValue`. • `BatchGetMultipleItems`: Ruft mehrere Elemente abhängig von den Werten von `partitionKey`, die in einer XML-Datei spezifiziert sind. Alle Elemente werden mit allen Elementattributen abgeholt. **Erforderliche Parameter:** `tableName`, `partitionKey`, `partitionType`, `partitionValue` (muss mit dem Element, das abgerufen werden soll übereinstimmen). Bei sortierbare Schlüssel zusätzlich `sortKey`, `sortType`, `sortKeyValue`. • `Delete`: Löscht Element oder Tabelle gemäß den konfigurierten Parametern. ⓘ Wenn `tableName` und die Primärschlüssel-Elemente angegeben sind, dann löscht die Operation `Deletedas` spezifische Element. Wenn nur `tableName` angegeben ist, dann löscht die Operation `Delete` die Tabelle. • `BatchDeleteMultipleItems`: Löscht mehrere Elemente in einem einzigen Aufruf. **Erforderliche Parameter:** `tableName` ⓘ **Erforderliche Dateien** Für diese Operation wird eine Input-XML-Datei benötigt, die die Elemente enthält. • `Query`: Ruft Teilmenge von Elementen innerhalb einer Tabelle ab. Für die verschiedenen Attribute werden Bedingungen angegeben. Weitere Informationen zur Input-Datei, die die Query aufruft, siehe Input-Abschnitt. **Erforderliche Parameter:** `tableName`, `partitionKey` ⓘ `partitionKey` ist obligatorisch und der Komparator ist immer "=". Wenn `sortKey` vorhanden ist, ist der Komparator zwischen `partitionKey` und `sortKey` "AND". **Erlaubte Operatoren:**=, >, <, >=, <=, !=, BETWEEN, begins_with

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.amazondynamodb.AmazonDynamoDBAdapter*: Hauptklasse (Standard)
`keyID`	Zugangsschlüssel-ID für Anfragen an AWS (Amazon Web Services) Die Zugangsdaten werden über Parameter an den Adapter gesendet. Dabei gibt es verschiedene Konfigurationsmöglichkeiten, siehe Zugangsdaten einrichten. **Mögliche Werte:** Beliebige Zeichenkette mit der Zugangsschlüssel-ID, z. B. *`AKIAIOSFODNN7EXAMPLE`*
`secretKey`	Geheimer Zugangsschlüssel für Anfragen an AWS Die Zugangsdaten werden über Parameter an den Adapter gesendet. Dabei gibt es verschiedene Konfigurationsmöglichkeiten, siehe Zugangsdaten einrichten. **Mögliche Werte:** Beliebige Zeichenkette mit der geheimen Zugangsschlüssel, z. B. *`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`*
`region`	Region der Domains **Mögliche Werte:** Beliebige Zeichenkette mit dem Namen der Region, z. B. *`us-east-2`* Buckets, die in einer bestimmten Region erstellt wurden, können nicht von einer anderen Region aus aufgerufen werden, siehe Regionen und Endpoints.

`tableName`	Tabellen-Name
	
	Mögliche Werte: Beliebige Zeichenkette mit dem Tabellen-Namen
	
	Folgende Namensregeln gelten für Tabellen-Namen:
	• Alle Namen müssen UTF-8-kodiert sein
	• Die Groß- und Kleinschreibung muss beachtet werden
	• Tabellen-Namen und Index-Namen müssen zwischen 3 und 255 Zeichen lang sein und dürfen nur folgende Zeichen enthalten:
	  • `a-z`
	  • `A-Z`
	  • `0-9`
	  • `_` (Unterstrich)
	  • `-` (Bindestrich)
	  • `.` (Punkt)
	• Attribut-Namen müssen zwischen 1 und 255 Zeichen lang sein

`partitionKey`, `partitionType` und `partitionKeyValue`	Elemente des Tabellen-Primärschlüssels
	
	Mögliche Werte:
	
	• `PartitionKey`: Name des Partitionsschlüssels
	• `PartitionType`: Partitionstyp
	  • **S**: Das Attribut ist eine Zeichenkette
	  • **N**: Das Attribut ist eine Zahl
	  • **B**: Das Attribut ist vom Typ Binär
	• `PartitionKeyValue`: Wert des Elements; Wird für die Operation `GetItem` benötigt

`sortKey`, `sortType` and `sortKeyValue`	Elemente des Tabellen-Primärschlüssels
	
	Mögliche Werte:
	
	• `SortKey`: Name des Sortierschlüssels
	• `SortType`: Sortiertyp
	  • **S**: Das Attribut ist eine Zeichenkette
	  • **N**: Das Attribut ist eine Zahl
	  • **B**: Das Attribut ist vom Typ Binär
	• `SortKeyValue`: Wert des Elements; Wird für die Operation `GetItem` benötigt

`readCapacityUnits`, `writeCapacitiyUnits`	Read/Write Capacity Units **Mögliche Werte:** • Integer Capacity Units
`scalableType`	Definiert den Skalierbarkeitstyp für `AutoScaling`. **Mögliche Werte:** • `ReadCapacity` • `WriteCapacity`
`autoScaling`	Legt fest, dass die Tabelle mithilfe von `AutoScaling` erstellt wird. Im Fall einer Überprüfung müssen alle Scaling-Parameter angegeben werden. **Mögliche Werte:** • *`true`* • *`false`*

`scalingMinCapacity`, `scalingMaxCapacity`, `scalingTargetValue`, `scalingInCooldown`, `scalingOutCooldown`	Definiert die Scaling-Parameter für `AutoScaling`. **Mögliche Werte:** • Weitere Informationen siehe AutoScaling.
`arnRole`	ARN der Rolle, die autorisiert ist, AutoScaling für die Tabellenressource anzuwenden. **Mögliche Werte:** • Weitere Informationen siehe Amazon Resource Names (ARN)
`consistentRead`	DynamoDB verwendet Eventually Consistent Reads, sofern nichts anderes angeben ist. Leseoperationen wie `GetItem` und `Query` liefern einen ConsistentRead-Parameter. Wenn dieser Parameter auf `true` gesetzt ist, verwendet DynamoDB während des Betriebs Strongly Consistent Reads.

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt
`0`	Operation `Delete`: • Das Element existiert nicht • Die Tabelle existiert nicht
`-1`	Bei der Ausführung des Adapters kam es zu einem Fehler

Input

Der Adapter erwartet eine vordefinierte XML-Struktur als Input. Die Struktur ist abhängig von der verwendeten Operation.

- `PutItem`-Operation:

Erwartete Struktur für die Operation PutItem

```
<?xml version="1.0" encoding="UTF-8"
<RootElement>
    <Item>
        <Attribute name="..Name of PartionKey.." type="..Type of Partition.."
value="..Value of PartitionKey.."/>
        <Attribute name="..Name of SortKey.." type="..Type of Sort.." value="..
Value of SortKey.."/>
        <Attribute name="..Attribute Name.." type="..Attribute Type.." value=".
.Attribute Value.."/>
    </Item>
</RootElement>
```

Für die Operation `PutItem` gibt es keine Einschränkungen für Elementnamen. Die Attribute `name`, `value` und `type` sind jedoch erforderlich.

- `BatchGetMultipleItems`-Operation:

Erwartete Struktur für die Operation BatchGetMultipleItems

```
<?xml version="1.0" encoding="UTF-8" ?>
<PrimaryKeyValues>
    <PrimaryKey partitionKey="2" SortKey="Carlos" />
    <PrimaryKey partitionKey="5" SortKey="Luis" />
</PrimaryKeyValues>
```

Primary Keys können für verschiedene Objekte spezifiziert werden. Wenn das Objekt mit einer Tabelle übereinstimmt, in der nur `partitionKey` definiert ist, darf nur `partitionKey` angegeben werden, andernfalls `partitionKey` und `SortKey`.

- `BatchPutMultipleItems`-Operation:

Erwartete Struktur für die Operation BatchPutMultipleitems

```
<?xml version="1.0" encoding="UTF-8" ?>
<RootElement>
    <Item>
        <Attribute name="ID" type="N" value="5" />
        <Attribute name="Name" type="S" value="Luis" />
        <Attribute name="Phone" type="N" value="0034666777888" />
        <Attribute name="Bonus" type="BOOL" value="true" />
        <Attribute name="Colours" type="LIST">
            <Attribute type="S" value="Red" />
            <Attribute type="S" value="Green" />
            <Attribute type="S" value="Blue" />
            <Attribute name="Cars" type="MAP">
                <Attribute5 name="Phone" type="N" value="637559681" />
                <Attribute6 name="Bonus" type="BOOL" value="true" />
                <Attribute7 name="Profession" type="S" value="Programmer" />
                <Attribute7 name="DocumentCode" type="B" value="qwrqfgdas122345
5sdfffr" />
            </Attribute>
        </Attribute>
        <Attribute name="Cars" type="MAP">
            <Attribute5 name="Phone" type="N" value="637559681" />
            <Attribute6 name="Bonus" type="BOOL" value="true" />
            <Attribute7 name="Profession" type="S" value="Programmer" />
            <Attribute7 name="DocumentCode" type="B" value="qwrqfgdas1223455sdf
ffr" />
        </Attribute>
    </Item>
    <Item>
        <Attribute1 name="ID" type="N" value="5" />
        <Attribute3 name="Name" type="S" value="Luis" />
        <Attribute4 name="Surname" type="S" value="Delgado" />
        <Attribute2 name="Address" type="S" value="Pirandello" />
        <Attribute5 name="Phone" type="N" value="637559681" />
        <Attribute6 name="Bonus" type="BOOL" value="true" />
        <Attribute7 name="Profession" type="S" value="Programmer" />
        <Attribute7 name="DocumentCode" type="B" value="qwrqfgdas1223455sdfffr"
/>
    </Item>
</RootElement>
```

Es können mehrere Items spezifiziert werden. In den Items können Attribute vom Typ Boolean (BOOL), String (S), Number (N), Map (MAP), List (LIST) und Binary (B) spezifiziert werden. Der Output dieser Operation ist dieselbe XML-Datei. Das kann für weitere Operationen nützlich sein.

- `BatchDeleteMultipleItems`-Operation:

Erwartete Struktur für die Operation BatchDeleteMultipleItems

```
<?xml version="1.0" encoding="UTF-8" ?>
<RootElement>
    <Item>
        <Attribute1 name="ID" type="N" value="7" />
        <Attribute7 name="Name" type="S" value="Luis" />
    </Item>
    <Item>
        <Attribute1 name="ID" type="N" value="6" />
        <Attribute7 name="Name" type="S" value="Luis" />
    </Item>
</RootElement>
```

In diesem Input werden die Primärschlüssel der Elemente angegeben, die gelöscht werden sollen. Wenn der Primärschlüssel nur aus dem `partitionKey` besteht, wird nur der `partionKey` benötigt. Wenn der Primärschlüssel aus `partitionKey` und `SortKey` besteht, werden beide Schlüssel benötigt, damit das zu löschende Element eindeutig ist. Der Output dieser Operation ist auch der Input.

- `Query`-Operation:

Erwartete Struktur für die Operation Query

```
<?xml version="1.0" encoding="UTF-8" ?>
<Query>
    <KeyCondition>ID=:param1</KeyCondition>
    <OtherFilters>Colors[3].Profession=:param3 or DocumentCode=:param2</
OtherFilters>
    <QueryParameters>
        <Parameter type="N" alias=":param1" value="5" />
        <Parameter type="B" alias=":param2" value="cXdycWZnZGFzMTIyMzQ1NXNkZmZm
cg==" />
        <Parameter type="S" alias=":param3" value="Programmer" />
    </QueryParameters>
</Query>
```

Das Elternelement `Query` enthält drei Elemente:

- `KeyCondition`: enthält nur den spezifischen `partitionKey` und falls vorhanden `SortKey`, die folgende Bedingungen erfüllen:
  - Der Komparator zwischen `partitionKey` und `SortKey` ist immer "AND".
  - `partitionKey` ist obligatorisch und wird immer mit "=" verwendet.
  - Wenn `SortKey` verwendet wird, können folgende Operatoren verwendet werden: =, >, <, >=, <=, !=, BETWEEN, begins_with
  - `partitionKey` und `SortKey` dürfen nicht mit für AWS DynamoDB reservierten Wörtern übereinstimmen.
- `OtherFilters`: spezifiziert Bedingungen für die restlichen Attribute, die nicht Teil des Primärschlüssels sind. Die betroffenen Attribute dürfen nicht mit für AWS Dynamo DB reservierten Wörtern übereinstimmen.
- `QueryParameters`: spezifiziert das Mapping zwischen den verwendeten Parametern und Ihren entsprechenden Werten. Der Typ jedes Parameters muss spezifiziert werden.

Output

Je nach Operation, gibt der Adapter Daten in verschiedenen XML-Strukturen aus:

- `GetItem`-Operation

Beispiel-Output für die Operation GetItem

```
<?xml version="1.0" encoding="UTF-8"
<Result>
    <Item partitionKey="ID" partitionKeyValue="2" SortKey="Name" SortKeyValue="
Carlos">
        <Attribute name="Address" type="S" value="Pirandello"/>
        <Attribute name="Phone" type="N" value="637559681"/>
        <Attribute name="Bonus" type="BOOL" value="true"/>
        <Attribute name="DocumentCode" type="B" value="qwrqfgdas1223455sdfffr"/
>
        <Attribute name="Surname" type="S" value="Delgado"/>
        <Attribute name="Profession" type="S" value="Programmer"/>
    </Item>
</Result>
```

- `BatchGetMultipleItems`-Operation:

Beispiel-Output für die Operation BatchGetMultipleItems

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
    <Item partitionKey="ID" partitionKeyValue="5" SortKey="Name" SortKeyValue="
Luis">
    <Attribute type="MAP" name="Cars">
            <Attribute type="N" name="Phone" value="637559681" />
            <Attribute type="BOOL" name="Bonus" value="true" />
            <Attribute type="B" name="DocumentCode" value="qwrqfgdas1223455sdff
fr" />
    <Attribute type="S" name="Profession" value="Programmer" />
    </Attribute>
        <Attribute type="N" name="Phone" value="34666777888" />
    <Attribute type="LIST" name="Colours">
            <Attribute type="S" value="Red" />
            <Attribute type="S" value="Green" />
    <Attribute type="S" value="Blue" />
            <Attribute type="MAP" name="Colours">
                <Attribute type="N" name="Phone" value="637559681" />
                <Attribute type="BOOL" name="Bonus" value="true" />
                <Attribute type="B" name="DocumentCode" value="qwrqfgdas1223455
sdfffr" />
                <Attribute type="S" name="Profession" value="Programmer" />
    </Attribute>
    </Attribute>
        <Attribute type="BOOL" name="Bonus" value="true" />
    </Item>
    <Item partitionKey="ID" partitionKeyValue="2" SortKey="Name" SortKeyValue="
Carlos">
        <Attribute type="S" name="Address" value="Pirandello" />
        <Attribute type="N" name="Phone" value="637559681" />
        <Attribute type="BOOL" name="Bonus" value="true" />
    <Attribute type="B" name="DocumentCode" value="qwrqfgdas1223455sdfffr" />
        <Attribute type="S" name="Surname" value="Delgado" />
        <Attribute type="S" name="Profession" value="Programmer" />
    </Item>
</Result>
```

Für jedes Element entsteht ein `Item`-Element mit Feldern für den Primärschlüssel und die dazugehörigen Werte. Innerhalb des `Item`-Elements findet sich eine Liste der dazugehörigen Attribute.

- `ListTables`-Operation

Beispiel-Output für die Operation ListTables

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
    <sampleItems />
</Result>
```

ⓘ Es wirt ein Elternelement `Result` geliefert, das ein Element für jede Tabelle enthält, die im Account deklariert ist.

- `Query`-Operation

Beispiel-Output für die Operation Query

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result hasMorePages="false">
    <Item>
        <Attribute value="{Phone=637559681, Bonus=true,
DocumentCode=[B@231dee4d, Profession=Programmer}" name="Cars" />
        <Attribute value="34666777888" name="Phone" />
        <Attribute value="[Red, Green, Blue, {Phone=637559681, Bonus=true,
DocumentCode=[B@5857d888, Profession=Programmer}]" name="Colours" />
    <Attribute value="true" name="Bonus" />
    <Attribute value="5" name="ID" />
    <Attribute value="Luis" name="Name" />
    </Item>
</Result>
```

ⓘ Es wird ein Elternelement `Result` geliefert. Das Attribut `hasMorePages` gibt an, ob das Ergebnis noch mehr Seiten hat, die angezeigt werden sollen oder nicht. Das ist hilfreich bei umfangreichen Antworten. Für jedes Element, das die Abfragebedingungen erfüllt, wirt ein `Item`-Element erzeugt, das alle Attribute enthält.

10.2.2 Apache Cassandra Connector

`Apache Cassadra Connector`: Verbindet sich mit einer Apache Cassandra Datenbank.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Execute`: Führt das im Input-Dokument angegebene Statement aus • `Terminate`: Schließt und entfernt das Connection Pooling mit allen offenen Verbindungen

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.apache.cassandra.ApacheCassandraConnectorAdapter*: Hauptklasse (Standard)
`host`	Host innerhalb des Clusters `Cluster cluster1 = Cluster.builder().addContactPoints("172.30.210.11","172.30.210.12").build();` Sie können mehrere IPs oder Host-Namen hinzufügen, die verschiedenen Servern entsprechen, die als ausführbare Datei von Apache Cassandra innerhalb des Moduls ausgeführt werden.
`port`	Native-Protocol-Port
`login`	Login, wenn das Cluster Zugangsdaten erfordert **Mögliche Werte:** Beliebige Zeichenkette mit dem Login, z. B. *`myLogin`*
`password`	Passwort, wenn das Cluster Zugangsdaten erfordert **Mögliche Werte:** Beliebige Zeichenkette mit dem Passwort, z. B. *`myPassword`*

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt
`-1`	Bei der Ausführung des Adapters kam es zu einem Fehler

Input

Der Adapter erwartet ein spezifisches XML-Dokument als Input:

Beispiel-Input

```
<RootElement>
    <Preparedstatement resulttag="result element name" timeZone="timezone">
        <Cql>
            select * from keyspace.table;
        </Cql>
        <Parameters>
            <parameter index="n" type="Cassandra_Type">Value</parameter>
        </Parameters>
    </Preparedstatement>
    <Preparedstatement resulttag="result 2 element name" timeZone="timezone 2">
       <Cql>
            INSERT INTO keyspace.table (id, lastname, firstname) VALUES (uuid(),
myLastname,'myFirstName');
        </Cql>
    </Preparedstatement>
    ....
</RootElement>
```

Das Input-XML-Dokument muss folgenden Regeln entsprechen:

Element / Attribut	Beschreibung
`resulttag`	*Erforderliches* Attribut des Elements `Preparedstatement`.
`timeZone`	*Optionales* Attribut des Elements `Preparedstatement`. Bestimmt die Zeitzone entsprechend zu `java.util.TimeZone` (z. B. *GMT+2*). Standardmäßig wird die UTC-Zeitzone für Datumsangaben verwendet.
`Cql`	Element innerhalb des Elements `Preparedstatement`. Enthält das CQL-Statement. Kann beliebig oft innerhalb von `Preparedstatement` verwendet werden. ⓘ Wenn innerhalb des CQL-Statements Zeichen verwendet werden, die in XML vorbelegt sind, dann muss das SQL-Statement mit einem CDATA-Block escaped werden.
`Parameters`	Element innerhalb des Elements `Preparedstatement`. Darf nur genau einmal innerhalb von `Preparedstatement` verwendet werden. Enthält beliebig viele Elemente `parameter` mit Parameterwerten.

Element / Attribut	Beschreibung

`parameter`	Element innerhalb des Elements `Parameters`. Kann beliebig oft innerhalb von `Parameters` verwendet werden.
	
	Attribute:
	
	• `index`: Definiert die Position, an der der Parameterwert das `?` innerhalb des CQL-Statements ersetzt.
	• `type`: Definiert den Parametertyp. Folgende Werte sind mit Apache Cassandra möglich:
	  • `TEXT`: UTF-8-kodierter Text
	  • `VARCHAR`: UTF-8-kodierter Text
	  • `ASCII`: US-ASCII-kodierter Text
	  • `BLOB`: Hexadezimal kodierte Binärdateien ohne Validierung
	  • `BOOLEAN`: Bool'sche Werte
	  • `BIGINT`: 64-Bit-Ganzzahl
	  • `COUNTER`: 64-Bit-Zählerwert
	  • `INT`: 32-Bit-Ganzzahl
	  • `SMALLINT`: 2-Byte-Ganzzahl
	  • `TINYINT`: 1-Byte-Ganzzahl
	  • `DATE`: Datum im Format `yyyy-mm-dd`
	  ⓘ Apacha Cassandra kodiert Datumsangaben als 32-Bit-Ganzzahl, die die Tage seit dem 1. Januar 1970 repräsentiert.
	  • `DECIMAL`: Dezimalzahl
	  • `DOUBLE`: 64-Bit-IEEE-Fließkommazahl
	  • `FLOAT`: 32-Bit-IEEE-Fließkommazahl
	  • `INET`: IP-Adresse im IPv4- oder IPv6-Format.
	  • `TIME`: Uhrzeit
	  ⓘ Apache Cassandra kodiert Zeitangaben als 64-Bit-Ganzzahl, die die Nanosekunden seit Mitternacht repräsentiert.
	  • `TIMESTAMP`: Datum und Uhrzeit. Das Format muss über das Attribut `format` angegeben werden, z. B.:
	  `<parameter index="n" type="TIMESTAMP" format="yyyy-MM-dd HH:mm:ss">2012-06-01 08:00:00</parameter>`
	  • `TIMEUUID`: Version 1 UUID
	  • `UUID`: UUID im Standard UUID-Format
	  • `VARINT`: Ganzzahl
	  • `MAP`: Array von Zeichen im JSON-Format
	  • `LIST`: Sammlung eines oder mehrerer geordneter Elemente, z. B.

Element / Attribut	Beschreibung

```
<parameter index="n" type="LIST">["tcodd@relational.com","ted.cod
d@relational.com"]</parameter>
```

ⓘ Die Verwendung der Anführungszeichen ist erforderlich.

- `SET`: Sammlung eines oder mehrerer Elemente, z. B.

```
<parameter index="n" type="SET">["Element 1","Element 1","Element
3"]</parameter>
```

ⓘ Die Verwendung der Anführungszeichen ist erforderlich.

- `TUPLE`: Gruppe von Komma-getrennten Feldern. Kann verschiedene Simple Types enthalten:
  - `keySpace`: Name des Key Space der Tabelle
  - `table`: Name der Tabelle
  - `column`: Name der Spalte

ⓘ Wenn das Tuple Date-, Time- oder Timestamp-Felder enthält sind folgende Attribute erforderlich:

- `formatWithoutTime`: Erforderlich, falls das Tuple Date-Felder enthält.
- `format`: Erforderlich, falls das Tuple Timestamp-Felder enthält.
- `formatTime`: Erforderlich, falls das Tuple Time-Felder enthält.

Beispiel Tuple

```
<parameter index="n" type="TUPLE" keySpace="videodb" table="video
s" column="tupleColl"
formatWithoutTime="yyyy-MM-dd"
format="yyyy-MM-dd HH:mm:ss"
formatTime="HH:mm:ss:SSS">1, fisrt tuple element, V1, 1.01,
14:25:16:555, 2012-06-01 08:00:00, 2018-04-03, 127.0.0.1, 5.678,
9e816244-f90a-4394-86a5-ddcf4fb9f7aa, true, 97, blobValue</
parameter>
```

Element / Attribut	Beschreibung
	Im obigen Beispiel wird das Tuple-Feld während der Tabellenerstellung wie folgt definiert: `tupleColl tuple<int, text, varchar, float, time, timestamp, date, inet, decimal, uuid, boolean, ascii, blob>`

Ausgabe

Die Ausgabe ist ein XML-Dokument mit den Fehlern und dem Ergebnis der Adapter-Ausführung. Je nach Operation gibt der Adapter Daten in verschiedenen XML-Strukturen aus:

- Batch-Operationen (`INSERT`, `UPDATE`, `DROP`, `CREATE`, ...)

  Beispiel Batch-Operationen

```
<Result>
    <resulttag_name> Operation executed successfully</resulttag_name>
</Result>
```

- SELECT

  Beispiel SELECT

```
<Result>
    <resulttag_name>
        <Object>
            <Value name="field_name" type="String|Number|
Boolean...">field_value</Value>
            ...
            <Object name="Map_field_name">
                <Value name="key_name" type="String|Number|
Boolean...">object_value</Value>
            </Object>
            <Array name="Colleciton_field_name">
                <Value type="String|Number|
Boolean...">value_of_the_array_at_this_position</Value>
                ...
            </Array>
            ...
        </Object>
    </resulttag_name>
</Result>
```

Wenn das Feld den Datentyp `MAP` hat, wird sein Wert durch einen Objektknoten repräsentiert. Wenn das Feld den Datentyp `LIST`, `SET` oder `TUPLE` hat, wird sein Wert durch einen Array-Knoten repräsentiert. Wenn das Feld ein Simple Type ist, wird sein Wert innerhalb des Value-Knotens repräsentiert.

Beispiel

- CREATE

Beispiel CREATE

```
<RootElement>
    <Preparedstatement resulttag="createTableVideos">
        <Cql>
            <![CDATA[CREATE TABLE videodb.videos (
                videoid uuid,
                videoname varchar,
                username varchar,
                description varchar,
                location map<varchar,varchar>,
                tags set<varchar>,
                tupleColl tuple<int, text, varchar, float, time, timestamp,
date, inet, decimal, uuid, boolean, ascii, blob>,
                upload_date timestamp,
                upload_date_noTime date,
                PRIMARY KEY (videoid));]]>
        </Cql>
    </Preparedstatement>
</RootElement>
```

- DROP

Beispiel DROP

```
<RootElement>
    <Preparedstatement resulttag="drop_Users">
        <Cql>
            <![CDATA[DROP TABLE IF EXISTS videodb.users;]]>
        </Cql>
    </Preparedstatement>
</RootElement>
```

- INSERT

Beispiel INSERT

```
<RootElement>
    <Preparedstatement resulttag="insertVideos" timeZone="UTC">
        <Cql>
            INSERT INTO videodb.videos (videoid, videoname, username,
description, location, tags, tupleColl, upload_date, upload_date_noTime)
            VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?);
        </Cql>
        <Parameters>
            <parameter index="0" type="UUID">99051fe9-6a9c-46c2-
b949-38ef78858dd0</parameter>
            <parameter index="1" type="VARCHAR">My funny cat</parameter>
            <parameter index="2" type="VARCHAR">tcodd</parameter>
            <parameter index="3" type="VARCHAR">My cat likes to play the piano!
So funny.</parameter>
            <parameter index="4" type="MAP">{"US":"/us/vid/
99/99051fe9-6a9c-46c2-b949-38ef78858dd0"}</parameter>
            <parameter index="5" type="SET" >["cats","piano","lol"]</parameter>
            <parameter index="6" type="TUPLE" keySpace="videodb" table="videos"
 column="tupleColl" formatWithoutTime="yyyy-MM-dd" format="yyyy-MM-dd HH:mm:ss"
 formatTime="HH:mm:ss:SSS">1, fisrt tuple element, V1, 1.01, 14:25:16:555,
2012-06-01 08:00:00, 2018-04-03, 127.0.0.1, 5.678, 9e816244-f90a-4394-86a5-
ddcf4fb9f7aa, true, 97, blobValue</parameter>
            <parameter index="7" type="TIMESTAMP" format="yyyy-MM-dd
HH:mm:ss">2012-06-01 08:45:32</parameter>
            <parameter index="8" type="DATE" formatWithoutTime="yyyy-MM-
dd">2012-06-01</parameter>
        </Parameters>
    </Preparedstatement>
</RootElement>
```

- UPDATE

Beispiel UPDATE

```
<rootElement>
    <Preparedstatement resulttag="add_email_in_list">
        <Cql>
            <![CDATA[ UPDATE videodb.users set email = ? + email where username
= ?]]>
        </Cql>
        <Parameters>
            <parameter type="LIST" index="0">["test1@mail.com","test0@mail.com"
]</parameter>
            <parameter type="TEXT" index="1">pmcfadin</parameter>
        </Parameters>
    </Preparedstatement>
</rootElement>
```

- SELECT

Beispiel SELECT

```
<mainNode>
    <Preparedstatement resulttag="select_NY_rides" timezone="UTC">
        <Cql>
            <![CDATA[SELECT * from NY.rides
                where vendor_id = ?
                and pickup_datetime = ?
                and rate_code = ?
                and total_amount = ?
                limit 10 ALLOW FILTERING;
            ]]>
        </Cql>
        <Parameters>
            <parameter index="0" type="TEXT">2</parameter>
            <parameter index="1" type="TIMESTAMP" format="yyyy-MM-dd
hh:mm:ss">2016-01-02 17:02:28</parameter>
            <parameter index="2" type="INT">1</parameter>
            <parameter index="3" type="DECIMAL">64.56</parameter>
        </Parameters>
    </Preparedstatement>
</mainNode>
```

- DELETE

Beispiel DELETE

```
<mainNode>
    <Preparedstatement resulttag="delete_video_one_record" timezone="UTC">
        <Cql>
            DELETE FROM videodb.videos WHERE videoid = ?;
        </Cql>
        <Parameters>
            <parameter index="0" type="UUID">7dacb4da-9155-4613-
b04e-8c5305a20342</parameter>
        </Parameters>
    </Preparedstatement>
</mainNode>
```

10.2.3 Google Cloud Storage BigTable Connector

`Google Cloud BigTable Connecotr`: Ermöglicht den Anschluss an Google Cloud BigTable, einen hochverfügbaren NoSQL-Datenspeicher, der die Arbeit der Datenbankverwaltung entlastet.
Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `CreateTable`: Erstellt eine Tabelle mit Tabellenname und Spaltenfamilienname **Erforderliche Parameter:** `tableName`, `columnFamilyName` • `ListTable`: Listet alle Tabellen für das Projekt und die Instanz auf **Erforderliche Parameter:** `tableName`, `columnFamilyName` • `PutRow`: Fügt Zeilen mit den Werten aus dem XML-Eingabedokument ein. Verwendet `columnName` und `value` für das `rowKey`-Attribut. **Erforderliche Parameter:** `tableName`, `columnFamilyName` • `GetRow`: Durchsucht die Tabelle, indem RowKey über Parameter übergeben wird und gibt das Ergebnis in einer Ausgabe-XML zurück. **Erforderliche Parameter:** `tableName`, `columnFamilyName` • `Scan`: Scannt die Tabelle und gibt eine Ausgabe-XML entsprechend der Parameter zurück. Der Scanvorgang filtert die Suche. **Erforderliche Parameter:** `tableName`, `columnFamilyName`, `columnName`, `startRow`, `stopRow`, `scanCached`, `scanResultSize`, `scanTimestamp`, `scanMaxVersions`, `scanCacheBlocks` • `EnableTable`: Aktiviert die Tabelle (Tabelle muss deaktiviert sein). **Erforderliche Parameter:** `tableName`, `columnFamilyName` • `DisableTable`: Deaktiviert die Tabelle (Tabelle muss aktiviert sein). **Erforderliche Parameter:** `tableName`, `columnFamilyName` • `Delete`: Löscht die Tabelle. Wenn die Parameter `rowKey` und `columnName` angegeben sind, wird nur die entsprechende Spalte in der gewählten Zeile gelöscht. **Erforderliche Parameter:** `tableName`, `columnFamilyName`, `columnName`, `rowKey`

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.google.cloud.bigtable.storage.GoogleCloudBigtableAdapter*: Hauptklasse (Standard)

`serviceAccountFile`	Pfad zur JSON-Datei, die den Service Account Key enthält, um sich gegenüber dem Google Cloud Storage zu authentifizieren. **Mögliche Werte:** • Systempfad: z. B.: `C:\X4\V_5.4\Server\X4DB\1\ExampleProcess\GoogleCloud\Config\serviceAccountFile.json` • X4DB-URL: z. B.: `x4db://1/ExampleProcess/GoogleCloud/Config/serviceAccountFile.json` • Xstore-URL: z. B.: `xstore://ExampleProcess/GoogleCloud/Config/serviceAccountFile.json`
`projectID`	ID des Projekts **Mögliche Werte:** • ID eines Google Cloud Projekts. Kann in der Google Cloud Konsole eingesehen werden.
`instanceID`	ID der Instanz **Mögliche Werte :** • ID einer Google Cloud Instanz. Kann in der Google Cloud Konsole eingesehen werden. Weitere Informationen, wie eine Instanz in Google BigTable erstellt werden kann, finden sich hier.
`tableName`	Name der Tabelle. Notwendiger Parameter für alle Operationen
`columnFamilyName`	Spaltenfamilienname. Gruppiert Spalten mit eindeutigen Namen.
`rowKey`	Zeilenschlüssel der Zeile. Identifikator um Zeilen einzufügen oder zu durchsuchen.
`columnName`	Spaltenname. Wird verwendet um mit der `Delete`-Operation eine bestimmte Spalte in einer Zeile zu löschen.
`startRow`	Spezifiziert eine bestimmte Zeile, bei der der Scanvorgang beginnt. **Mögliche Werte:** Zeilenschlüssel `rowKey`
`stopRow`	Spezifiziert eine bestimmte Zeile, bei der der Scanvorgang endet. Die Zeile selbst wird nicht mehr in den Scan-Ergebnissen angezeigt. **Mögliche Werte:** Zeilenschlüssel `rowKey`

`scanCached`	Ändert das Scanner-Caching für diesen Scan. Wenn der Parameter nicht gesetzt ist, wird der Caching-Wert der Hosting-Tabelle verwendet. **Mögliche Werte:** Ganzzahl
`scanResultSize`	Legt die maximale Anzahl der Scan-Ergebnisse fest. **Mögliche Werte:** Ganzzahl ⓘ Wenn die beiden Parameter `scanCached` und `scanResultSize` gesetzt sind, werden einzelne Server Requests entweder durch die Anzahl der Zeilen oder die maximale Ergebnisgröße begrenzt, je nachdem, welches Limit zuerst eintritt.
`scanTimestamp`	Filtert nach Spalten mit einem bestimmten Zeitstempel. **Mögliche Werte:** Ganzzahl
`scanMaxVersions`	Begrenzt die Anzahl Versionen jeder Spalte, die zurückgegeben werden sollen. **Mögliche Werte:** Ganzzahl
`scanCacheBlocks`	Serverseitiges Caching für diesen Scan explizit deaktivieren. **Mögliche Werte:** `true` / `false`

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`0`	Die Adapter-Operation wurde nicht vollständig ausgeführt, es ist aber kein Fehler aufgetreten.
`-1`	Bei der Ausführung des Adapters ist ein Fehler aufgetreten.

Input

- Der Adapter erwartet für die `PutRow`-Operation XML in folgender Struktur:

```
<ListRows>
    <row rowKey="IT employee">
        <rowAttribute1 columnName="ID" value="00001" />
        <rowAttribute2 columnName="Address" value="Karlsruhe" />
        <rowAttribute3 columnName="Name" value="Paolo" />
        <rowAttribute4 columnName="Surname" value="Green" />
        <rowAttribute5 columnName="Phone" value="552-968-369" />
        <rowAttribute6 columnName="Bonus" value="Yes" />
        <rowAttribute7 columnName="Profession" value="System Administrator" />
    </row>
</ListRows>
```

> Elementnamen können frei gewählt werden, allerdings sind die Attribute `rowKey`, `columnName` und `value` erforderlich.

Output

Der Adapter liefert abhängig von der ausgeführten Operation unterschiedliche Arten von Dokumenten zurück.

- `GetRow`-Operation ausführen:

```
<Result>
    <Row rowKey="DEV employee">
        <RowColumn ColumnName="Address" value="Ettlingen" />
<RowColumn ColumnName="Frameworks" value="Java-Hibernate" />
        <RowColumn ColumnName="ID" value="00003" />
        <RowColumn ColumnName="Name" value="Lorena" />
    </Row>
</Result>
```

- `Scan`-Operation ausführen:

```
<Column FamilyName="FamilyNamePrimary">
    <Row rowKey="CLean employee">
        <Attribute ColumnName="Address" value="Ettlingen" timeStamp="1544603282
081" />
        <Attribute ColumnName="ID" value="00005" timeStamp="1544603282081" />
    </Row>
    <Row rowKey="DEV employee">
        <Attribute ColumnName="Address" value="Ettlingen" timeStamp="1544603280
340" />
        <Attribute ColumnName="Frameworks" value="Java-Hibernate" timeStamp="15
44603280340" />
        <Attribute ColumnName="ID" value="00003" timeStamp="1544603280340" />
        <Attribute ColumnName="Name" value="Lorena" timeStamp="1544603280340" /
>
    </Row>
    <Row rowKey="HR employee">
        <Attribute ColumnName="ID" value="00002" timeStamp="1544603278048" />
        <Attribute ColumnName="Name" value="Francis Ernesto" timeStamp="1544603
278048" />
        <Attribute ColumnName="Surname" value="Underwood" timeStamp="1544603278
048" />
    </Row>
    <Row rowKey="IT employee">
        <Attribute ColumnName="Address" value="Karlsruhe" timeStamp="1544603276
625" />
        <Attribute ColumnName="Bonus" value="Yes" timeStamp="1544603276625" />
        <Attribute ColumnName="ID" value="00001" timeStamp="1544603276625" />
        <Attribute ColumnName="Name" value="Paolo" timeStamp="1544603276625" />
        <Attribute ColumnName="Phone" value="552-968-369" timeStamp="1544603276
625" />
    <Attribute ColumnName="Profession" value="System Administrator" timeStamp="
1544603276625" />
        <Attribute ColumnName="Surname" value="Green" timeStamp="1544603276625"
/>
    </Row>
    <Row rowKey="Marketing employee">
        <Attribute ColumnName="ID" value="00004" timeStamp="1544603281175" />
    <Attribute ColumnName="Name" value="Petra" timeStamp="1544603281175" />
    </Row>
    <Row rowKey="Security employee">
        <Attribute ColumnName="Address" value="Ettlingen/Build1" timeStamp="154
4603283155" />
        <Attribute ColumnName="ID" value="00006" timeStamp="1544603283155" />
        <Attribute ColumnName="Name" value="Robert" timeStamp="1544603283155" /
>
        <Attribute ColumnName="Permisions" value="Day-Night" timeStamp="1544603
283155" />
    </Row>
</Column>
```

- `ListTable`-Operation ausführen:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
    <ListTables>
    <Table name="TestTable" />
    </ListTables>
</Result>
```

10.2.4 Microsoft Azure Data Lake Gen 1 Storage Connector

`Microsoft Azure Data Lake Gen 1 Storage Connector`: Ermöglicht die Kommunikation mit Microsoft Azure Date Lakes der ersten Generation (basiert auf Apache Hadoop).

Eigenschaften

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • `AppendFile`: Datei zu einer bereits vorhandenen Datei hinzufügen. • `ConcatenateFiles`: Eine Liste von Dateien zu einer Zieldatei zusammenführen. Nach der erfolgreichen Zusammenführung werden die Quelldateien gelöscht. Ist die Zieldatei noch nicht vorhanden, wird diese automatisch angelegt. Andernfalls wird der Inhalt zusammengeführt. • `CreateFolder`: Ordner erstellen. • `DeleteFile`: Datei löschen. • `DeleteFolder`: Ordner uns seinen gesamten Inhalt rekursiv löschen. • `EnumerateDirectory`: Alle Dateien und Ordner aus einem bestimmten Ordnerpfad auflisten. • `GetMetadata`: Metadaten einer Datei oder eines Ordners auslesen. • `ReadFile`: Datei lesen. • `RenameFile`: Datei umbenennen. Wenn die Zieldatei bereits existiert (eine Datei mit demselben Namen), wird die Quelldatei nur dann umbenannt, wenn der Parameter `overwrite` aktiviert ist. • `RenameFolder`: Ordner umbenennen. Wenn der Zielordner bereits existiert, wird der Quellordner in den Zielordner verschoben. • `SetPermission`: Berechtigungen für eine Datei oder einen Ordner erteilen. • `UploadFile`: Neue Datei anlegen.

Parameters

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.microsoft.azure.datalake.MicrosoftAzureDataLakeStoreConnector*: Hauptklasse (Standard)

`authEndpoint`	*Pflichtparameter*: OAuth 2.0 Token-Endpunkt (v1) **Mögliche Werte:** Zeichenkette mit einem gültigen Endpunkt, z. B. *`https://login.microsoftonline.com/<DirectoryID>/oauth2/token`*
`clientID`	*Pflichtparameter*: Client-ID der "Azure Active Directory"-Anwendung. **Mögliche Werte:** Zeichenkette mit der Client-ID, z. B. *`10925xxx-1111-4444-bd3d-82165c0d3841`*
`clientSecret`	*Pflichtparameter*: Geheimer Clientschlüssel der "Azure Active Directory"-Anwendung. **Mögliche Werte:** Zeichenkette mit dem geheimen Clientschlüssel.
`accountName`	*Pflichtparameter*: Voll qualifizierter Data Lake Storage Account-Name (FQDN). **Mögliche Werte:** Zeichenkette mit dem FQDN, z. B. *`<your_account_name>.azuredatalakestore.net`*
`fileName`	Dateiname • Dieser Parameter ist für die Operationen `AppendFile`, `ConcatenateFiles`, `DeleteFile`, `ReadFile`, `RenameFile` und `UploadFile` erforderlich. • Dieser Parameter ist optional für die Operationen `GetMetadata` und `SetPermission`. **Mögliche Werte:** Zeichenkette mit dem Dateinamen, z. B. *`inputFile.txt`* oder *`root/subfolder/inputFile.txt`* Die Zeichenkette darf sowohl den vollständigen Dateipfad (*`root/subfolder/inputFile.txt`*) als auch den Dateinamen (`fileName` *`inputFile.txt`*) und Ordnernamen (`folderPath` *`root/subfolder`*) enthalten.
`folderPath`	Ordnerpfad; Kann mit einem Schrägstrich beginnen und enden. **Mögliche Werte:** • */*: Wurzelordner • Beliebiger gültiger Pfad, z. B. *`root/subFolder/.../lastFolder`*

`overwrite`	Angabe, ob vorhandene Dateien überschrieben werden sollen. ⓘ Dieser Parameter ist nur für die Operationen `UploadFile` und `RenameFile` gültig. **Mögliche Werte:** • `true`: Vorhandene Datei (in `fileName`) angegeben) mit neuem Inhalt überschreiben • `false`: Für die Operation `UploadFile` gibt der Adapter einen Fehler aus, wenn die Datei bereits existiert. Für die Operation `RenameFile` gibt der Adapter den Status `1` aus, benennt die Quelldatei jedoch nicht um.
`listFilesToConcatenate`	Name der Dateien, die zusammengeführt werden sollen getrennt durch einen Doppelpunkt. ⓘ Dieser Parameter ist für die Operation `ConcatenateFiles` notwendig. **Mögliche Werte:** Vorhandene Dateien (vollständige Pfadangabe) durch Doppelpunkt getrennt, z.B. *`files/File1.txt:files/File2.txt:otherPath/File3.txt`*
`renamedObject`	Neuer Pfadname der umzubenennenden Datei oder des Ordners. ⓘ Dieser Parameter ist für die Operationen `RenameFile` und `RenameFolder` notwendig. **Mögliche Werte:** Pfadname, z.*B.* *`files/newFileName.txt`*
`octalPermissions`	Zu setzende Berechtigungen in Unix-Octal-Format. Für mehr Informationen zu Unix-Dateirechten, siehe auch https://www.tutorialspoint.com/unix/unix-file-permission.htm. *Beispiel*: *`770`* → *`7`* Eigener Benutzer mit Lese-, Schreib- und Ausführ-Berechtigung; *`7`* Gruppe mit Lese-, Schreib- und Ausführ-Berechtigung; *`0`* Andere Benutzer ohne Berechtigungen. ⓘ Dieser Parameter ist für die Operation `SetPermission` notwendig.

`maxEntriesToRetrieve`	Maximale Anzahl von Einträgen, die abgerufen werden sollen. ⓘ Dieser Parameter ist optional und wird ignoriert, wenn der Parameter `entriesRecursively` aktiviert ist. Hilfreich für die Operation `EnumerateDirectory`. ⚠ Beachten Sie, dass der Server die Anzahl der abgerufenen Einträge auf eine kleinere Zahl als die angegebene Zahl begrenzen kann. **Mögliche Werte:** Ganze Zahl größer `0`.
`startAfterPathName`	Datei- oder Ordnername, nach dem die Aufzählung beginnen soll. ⓘ Dieser Parameter ist optional and wird ignoriert, wenn der Parameter `entriesRecursively` aktiviert ist. Hilfreich für die Operation `EnumerateDirectory`. **Mögliche Werte:** Zeichenkette mit dem Pfadnamen der Datei oder des Ordners.
`entriesRecursively`	Ermöglicht es, Einträge rekursiv zu extrahieren. Wenn der Parameter aktiviert ist, werden `maxEntriesToRetrieve` und `startAfterPathName` ignoriert. ⓘ Dieser Parameter ist optional und für die Operation `EnumerateDirectory` hilfreich. **Mögliche Werte:** • `true`: Alle Einträge des angegebenen Ordnerpfads werden rekursiv abgerufen. • `false`: Alle Einträge des angegebenen Ordnerpfads werden abgerufen, sodass nur direkte Kind-Elemente abgerufen werden.

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt
`-1`	Bei der Ausführung kam es zu einem Fehler

Input

Für die Operationen `UploadFile` und `AppendFile` erwartet der Adapter ein beliebiges Input-Dokument. Alle anderen Operationen benötigen kein Input.

Ausgabe

Je nach gewählter Operation gibt der Adapter verschiedene Dokumente aus:

AppendFile

- Wurde die Operation erfolgreich ausgeführt (Status 1), gibt der Adapter das Input-Dokument zurück.
- Im Falle eines Fehlers (Status -1), gibt der Adapter ein XML-Dokument mit der Fehlerbeschreibung zurück, z. B.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>com.microsoft.azure.datalake.store.ADLException: Error appending to file /ivan/subfolder22/input.txt Operation APPEND failed with HTTP404 :
    FileNotFoundException Last encountered exception thrown after 1 tries. [HTTP404(FileNotFoundException)] [ServerRequestId:7eacc9bc-a0d4-48c0-
    b15a-678156cb10c5]</Error>
```

CreateFolder

- Wurde die Operation erfolgreich ausgeführt (Status 1), gibt der Adapter das Input-Dokument zurück – falls ein Input-Dokument zugewiesen wurde.
- Im Falle eines Fehlers (Status -1), gibt der Adapter ein XML-Dokument mit der Fehlerbeschreibung zurück, z. B.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>com.microsoft.azure.datalake.store.ADLException: Error creating directory path is null Last encountered exception thrown after 1 tries. [HTTP0
    (null)] [ServerRequestId:null]</Error>
```

ConcatenateFiles

- Wurde die Operation erfolgreich ausgeführt (Status 1), gibt der Adapter das Input-Dokument zurück.
- Im Falle eines Fehlers (Status -1), gibt der Adapter ein XML-Dokument mit der Fehlerbeschreibung zurück, z. B.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>com.microsoft.azure.datalake.store.ADLException: Error concatenating files into /files/File1_File2.txt concat() source list contains a file more
    than once: Last encountered exception thrown after 1 tries. [ServerRequestId:null]</Error>
```

DeleteFile

- Wurde die Operation erfolgreich ausgeführt (Status 1), gibt der Adapter das Input-Dokument zurück – falls ein Input-Dokument zugewiesen wurde.
- Im Falle eines Fehlers (Status -1), gibt der Adapter ein XML-Dokument mit der Fehlerbeschreibung zurück, z. B.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>com.microsoft.azure.datalake.store.ADLException: Error deleting directory path is null Last encountered exception thrown after 1 tries. [HTTP0
    (null)] [ServerRequestId:null]</Error>
```

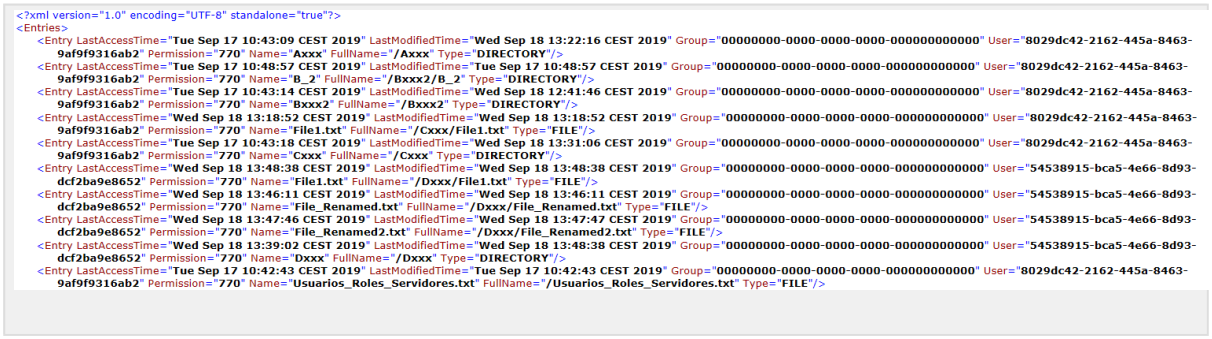
DeleteFolder

- Wurde die Operation erfolgreich ausgeführt (Status 1), gibt der Adapter das Input-Dokument zurück – falls ein Input-Dokument zugewiesen wurde.
- Im Falle eines Fehlers (Status -1), gibt der Adapter ein XML-Dokument mit der Fehlerbeschreibung zurück, z. B.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>com.microsoft.azure.datalake.store.ADLException: Error deleting directory tree path is null Last encountered exception thrown after 1 tries.
    [HTTP0(null)] [ServerRequestId:null]</Error>
```

`EnumerateDirectory`

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt der Adapter ein XML-Dokument mit einer Liste aller Dateien und Ordner aus, z. B.:



```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<Entries>
    <Entry LastAccessTime="Tue Sep 17 10:43:09 CEST 2019" LastModifiedTime="Wed Sep 18 13:22:16 CEST 2019" Group="00000000-0000-0000-0000-000000000000" User="8029dc42-2162-445a-8463-
        9af9f9316ab2" Permission="770" Name="Axxx" FullName="/Axxx" Type="DIRECTORY"/>
    <Entry LastAccessTime="Tue Sep 17 10:48:57 CEST 2019" LastModifiedTime="Tue Sep 17 10:48:57 CEST 2019" Group="00000000-0000-0000-0000-000000000000" User="8029dc42-2162-445a-8463-
        9af9f9316ab2" Permission="770" Name="B_2" FullName="/Bxxx2/B_2" Type="DIRECTORY"/>
    <Entry LastAccessTime="Tue Sep 17 10:43:14 CEST 2019" LastModifiedTime="Wed Sep 18 12:41:46 CEST 2019" Group="00000000-0000-0000-0000-000000000000" User="8029dc42-2162-445a-8463-
        9af9f9316ab2" Permission="770" Name="Bxxx2" FullName="/Bxxx2" Type="DIRECTORY"/>
    <Entry LastAccessTime="Wed Sep 18 13:18:52 CEST 2019" LastModifiedTime="Wed Sep 18 13:18:52 CEST 2019" Group="00000000-0000-0000-0000-000000000000" User="8029dc42-2162-445a-8463-
        9af9f9316ab2" Permission="770" Name="File1.txt" FullName="/Cxxx/File1.txt" Type="FILE"/>
    <Entry LastAccessTime="Tue Sep 17 10:43:18 CEST 2019" LastModifiedTime="Wed Sep 18 13:31:06 CEST 2019" Group="00000000-0000-0000-0000-000000000000" User="8029dc42-2162-445a-8463-
        9af9f9316ab2" Permission="770" Name="Cxxx" FullName="/Cxxx" Type="DIRECTORY"/>
    <Entry LastAccessTime="Wed Sep 18 13:48:38 CEST 2019" LastModifiedTime="Wed Sep 18 13:48:38 CEST 2019" Group="00000000-0000-0000-0000-000000000000" User="54538915-bca5-4e66-8d93-
        dcf2ba9e8652" Permission="770" Name="File1.txt" FullName="/Dxxx/File1.txt" Type="FILE"/>
    <Entry LastAccessTime="Wed Sep 18 13:46:11 CEST 2019" LastModifiedTime="Wed Sep 18 13:46:11 CEST 2019" Group="00000000-0000-0000-0000-000000000000" User="54538915-bca5-4e66-8d93-
        dcf2ba9e8652" Permission="770" Name="File_Renamed.txt" FullName="/Dxxx/File_Renamed.txt" Type="FILE"/>
    <Entry LastAccessTime="Wed Sep 18 13:47:46 CEST 2019" LastModifiedTime="Wed Sep 18 13:47:47 CEST 2019" Group="00000000-0000-0000-0000-000000000000" User="54538915-bca5-4e66-8d93-
        dcf2ba9e8652" Permission="770" Name="File_Renamed2.txt" FullName="/Dxxx/File_Renamed2.txt" Type="FILE"/>
    <Entry LastAccessTime="Wed Sep 18 13:39:02 CEST 2019" LastModifiedTime="Wed Sep 18 13:48:38 CEST 2019" Group="00000000-0000-0000-0000-000000000000" User="54538915-bca5-4e66-8d93-
        dcf2ba9e8652" Permission="770" Name="Dxxx" FullName="/Dxxx" Type="DIRECTORY"/>
    <Entry LastAccessTime="Tue Sep 17 10:42:43 CEST 2019" LastModifiedTime="Tue Sep 17 10:42:43 CEST 2019" Group="00000000-0000-0000-0000-000000000000" User="8029dc42-2162-445a-8463-
        9af9f9316ab2" Permission="770" Name="Usuarios_Roles_Servidores.txt" FullName="/Usuarios_Roles_Servidores.txt" Type="FILE"/>
```

> Jedes `Entry`-Element entspricht einer Datei oder einem Ordner. Folgende Attribute beschreiben das `Entry`-Element:
>
> - `LastAccessTime`: Zeitpunkt des letzten Zugriffs
> - `LastModifiedTime`: Zeitpunkt der letzten Änderung
> - `Group`: ID der Gruppe, die die Datei oder den Ordner besitzt
> - `User`: ID des Benutzers, der die Datei oder den Ordner besitzt
> - `Permission`: Berechtigungszeichenfolge für die Datei in UNIX-Syntax
> - `Name`: Dateiname
> - `FullName`: Vollständiger Datei- oder Ordnerpfad

- Im Falle eines Fehlers (Status `-1`), gibt der Adapter ein XML-Dokument mit der Fehlerbeschreibung zurück, z. B.:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>The folderPath parameter must be informed</Error>
```

`GetMetadata`

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt der Adapter ein XML-Dokument mit den Metadaten der Datei oder des Ordners aus.

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<Metadata>
    <Entry LastAccessTime="Wed Sep 18 09:07:45 CEST 2019" LastModifiedTime="Wed Sep 18 09:07:45 CEST 2019" Group="00000000-0000-0000-0000-
        000000000000" User="54538915-bca5-4e66-8d93-dcf2ba9e8652" Permission="770" Name="File1_File2.txt" FullName="/files/File1_File2.txt"
        Type="FILE"/>
</Metadata>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<Metadata>
    <Entry LastAccessTime="Tue Sep 17 10:14:18 CEST 2019" LastModifiedTime="Wed Sep 18 09:07:45 CEST 2019" Group="00000000-0000-0000-0000-
        000000000000" User="8029dc42-2162-445a-8463-9af9f9316ab2" Permission="770" Name="files" FullName="files" Type="DIRECTORY"/>
</Metadata>
```

> Das Wurzelelement `Metadata` enthält ein `Entry`-Element mit folgenden Attributen:
> - `LastAccessTime`: Zeitpunkt des letzten Zugriffs
> - `LastModifiedTime`: Zeitpunkt der letzten Änderung
> - `Group`: ID der Gruppe, die die Datei oder den Ordner besitzt
> - `User`: ID des Benutzers, der die Datei oder den Ordner besitzt
> - `Permission`: Berechtigungszeichenfolge für die Datei oder den Ordner in UNIX-Stil
> - `Name`: Datei- oder Ordnername
> - `FullName`: Vollständiger Datei- oder Ordnerpfad

- Im Falle eines Fehlers (Status `-1`), gibt der Adapter ein XML-Dokument mit der Fehlerbeschreibung zurück, z. B.:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>com.microsoft.azure.datalake.store.ADLException: Error getting info for file File1_File2.txt Operation GETFILESTATUS failed with HTTP404 :
    FileNotFoundException Last encountered exception thrown after 1 tries. [HTTP404(FileNotFoundException)] [ServerRequestId:c547c88f-ed41-
    40a4-bcc9-5f319fd83da7]</Error>
```

`ReadFile`

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt der Adapter die eingelesene Datei aus
- Im Falle eines Fehlers (Status `-1`), gibt der Adapter ein XML-Dokument mit der Fehlerbeschreibung zurück, z. B.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>com.microsoft.azure.datalake.store.ADLException: Error getting info for file path is null Last encountered exception thrown after 1 tries. [HTTP0
    (null)] [ServerRequestId:null]</Error>
```

`RenameFile`

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt der Adapter das Input-Dokument zurück – falls ein Input-Dokument zugewiesen wurde.
- Im Falle eines Fehlers (Status `-1`), gibt der Adapter ein XML-Dokument mit der Fehlerbeschreibung zurück, z. B.:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>The 'renamedObject' parameter must be set for this operation</Error>
```

`RenameFolder`

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt der Adapter das Input-Dokument zurück – falls ein Input-Dokument zugewiesen wurde.
- Im Falle eines Fehlers (Status `-1`), gibt der Adapter ein XML-Dokument mit der Fehlerbeschreibung zurück, z. B.:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>The 'renamedObject' parameter must be set for this operation</Error>
```

`SetPermission`

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt der Adapter das Input-Dokument zurück – falls ein Input-Dokument zugewiesen wurde.
- Im Falle eines Fehlers (Status `-1`), gibt der Adapter ein XML-Dokument mit der Fehlerbeschreibung zurück, z. B.:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>com.microsoft.azure.datalake.store.ADLException: Error setting times for /folderA/folderA_2/folderB2/New Text Document.txt Specified permissions are not valid Octal
    Permissions: 66055 Last encountered exception thrown after 1 tries. [ServerRequestId:null]</Error>
```

`UploadFile`

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt der Adapter die hochgeladene Datei aus
- Im Falle eines Fehlers (Status `-1`), gibt der Adapter ein XML-Dokument mit der Fehlerbeschreibung zurück, z. B.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>com.microsoft.azure.datalake.store.ADLException: Error creating file path is null Last encountered exception thrown after 1 tries. [HTTP0(null)]
    [ServerRequestId:null]</Error>
```

> Ist der Parameter `folderPath` leer, wird die Datei im Wurzelordner des Benutzerkontos angelegt. Existiert der Pfad nicht, der im Parameter `folderPath` gesetzt wurde, wird dieser angelegt.

10.3 Cloud

- Amazon S3 Simple Storage Connector
- Amazon SimpleDB Connector
- Dropbox Connector
- Google Cloud Storage Objects Connector
- Google Drive Connector
- Microsoft Azure Blob Storage Connector
- Microsoft Azure Table Storage Connector

10.3.1 Amazon S3 Simple Storage Connector

`Amazon S3 Simple Storage Connector`: Speichert Daten in Amazon S3 Blob Storage und ermöglicht die Verwendung der X4 BPMS, um Daten in Cloud-Umgebungen zu persistieren.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `CreateBucket`: Legt ein Bucket an. Für die Operation `CreateBucket` können verschiedene Konfigurationen vorgenommen werden. Ist beispielsweise der Parameter `bucketVersion` auf *ENABLED* gesetzt, wird die Bucket-Versionierung aktiviert. Andernfalls ist die Versionierung deaktiviert. Ist der Parameter `publicAccess` aktiviert, muss über den Parameter `publicPermission` eine Berechtigung definiert werden. **Erforderliche Parameter:** `bucket`, `bucketVersion`, `publicAccess`, `publicPermission` • `ChangeBucketVersion`: Ändert die Versionierungs-Konfiguration des Buckets. Neue Buckets sind standardmäßig im `OFF`-Status. Sobald die Versionierung aktiviert wird, kann der Status nicht mehr auf `OFF` zurückgesetzt werden. **Erforderliche Parameter:** `bucket`, `bucketVersion` • `GetAccessControlList`: Listet die Rechte und Berechtigungen für ein Bucket oder eine Datei auf. Sind die Parameter `bucket` und `file` angegeben, werden die Rechte und Berechtigungen für die Datei ausgegeben. Ist nur `bucket` angegeben, werden die Rechte und Berechtigungen für den Bucket ausgegeben. **Erforderliche Parameter:** `bucket`, `file` • `AddGrantsPermission`: Fügt einem Bucket oder einer Datei Rechte mit Berechtigungen hinzu . Ist der Parameter `publicAccess` aktiviert, wird ein öffentlich zugänglicher Bucket oder eine Datei angelegt mit den Berechtigungen, die im Parameter `publicPermission` definiert wurden. Andernfalls werden Informationen aus dem Input genommen. Sind die Parameter `bucket` und `file` angegeben, werden die Rechte und Berechtigungen zur Datei hinzugefügt. Ist nur `bucket` angegeben, werden die Rechte und Berechtigungen dem Bucket hinzugefügt. **Erforderliche Parameter:** `bucket`, `file` • `RevokeGrantsPermission`: Widerruft alle Berechtigungen. Sind die Parameter `bucket` und `file` angegeben, werden die Rechte und Berechtigungen für die Datei widerrufen. Ist nur `bucket` angegeben, werden die Rechte und Berechtigungen für den Bucket widerrufen. **Erforderliche Parameter:** `bucket`, `file` • `RevokePublicPermission`: Widerruft die Berechtigungen für den öffentlichen Zugang. Sind die Parameter `bucket` und `file` angegeben, werden die Berechtigungen für den öffentlichen Zugang auf eine Datei widerrufen. Ist nur `bucket` angegeben, werden die Berechtigungen für den öffentlichen Zugang auf den Bucket widerrufen. **Erforderliche Parameter:** `bucket`, `file` • `AddBucketPolicy`: Fügt eine Bucket-Richtlinie zu einem angegebenen Bucket hinzu. Unter folgenden Links finden Sie Zusatzinformationen, wie Beispiele für Bucket-Richtlinien, Berechtigungen in einer Richtlinien und Bedingungen in Richtlinien. **Erforderliche Parameter:** `bucket`

- `GetBucketPolicy`: Ruft die Bucket-Richtlinie auf.
  Erforderliche Parameter: `bucket`

- `RemoveBucketPolicy`: Löscht die Richtlinie für den angegebenen Bucket.
  Erforderliche Parameter: `bucket`

- `Search`: Sucht nach Informationen im AmazonS3 Simple Storage Konto und optional auch noch nach Objekt-Metadaten.
  Ist der Parameter `includeObjectsMetadata` ausgewählt, werden Metadaten-Informationen zu jedem Objekt der Standard-Antwort hinzugefügt.
  Erforderliche Parameter: `file`, `includeObjectsMetadata`

- `Upload`: Lädt Dateien zum Amazon S3 Simple Storage Konto hoch.
  Legen Sie Ordnerstrukturen mit "/" an, z. B. `Lib/Document/doc.txt`.
  Erforderliche Parameter: `bucket`, `file`, `overwrite`, `path`

- `Download`: Lädt Dateien vom Amazon S3 Simple Storage Konto herunter.
  Erforderliche Parameter: `bucket`, `file`

- `Copy`: Kopiert Dateien in den Ziel-Bucket.
  Um eine Datei zu kopieren, ist eine Berechtigung für das Ziel-Bucket erforderlich.
  Erforderliche Parameter: `bucket`, `file`, `targetBucket`, `targetFile`

- `Move`: Verschiebt Dateien in den Ziel-Bucket .
  Um eine Datei zu verschieben, ist eine Berechtigung für das Ziel-Bucket erforderlich.
  Erforderliche Parameter: `bucket`, `file`, `targetBucket`, `targetFile`

- `Rename`: Benennt Dateien in den Ziel-Bucket um.

- `Delete` : Löscht Dateien oder Buckets.
  Sind `bucket` und `file` angegeben, wird die angegebene Datei gelöscht. Ist nur `bucket` angegeben, werden alle Dateien im Bucket und anschließend der Bucket selbst gelöscht. Ist die Bucket-Versionierung eingeschaltet, werden zuerst die Dateiversionen und anschließend der Bucket gelöscht.

- `DeleteMultiFiles`: Löscht mehrere Dateien.
  Erforderliche Parameter: `bucket`, `file`

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.amazons3.AmazonS3SimpleStorageAdapter*: Hauptklasse (Standard)

`keyID`	Zugangsschlüssel-ID für Anfragen an AWS (Amazon Web Services) Die Zugangsdaten werden über Parameter an den Adapter gesendet. Dabei gibt es verschiedene Konfigurationsmöglichkeiten, siehe Zugangsdaten einrichten. **Mögliche Werte:** Beliebige Zeichenkette mit der Zugangsschlüssel-ID, z. B. *AKIAIOSFODNN7EXAMPLE*
`secretKey`	Geheimer Zugangsschlüssel für Anfragen an AWS Die Zugangsdaten werden über Parameter an den Adapter gesendet. Dabei gibt es verschiedene Konfigurationsmöglichkeiten, siehe Zugangsdaten einrichten . **Mögliche Werte:** Beliebige Zeichenkette mit der geheimen Zugangsschlüssel, z. B. *wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY*
`region`	Bucket-Region **Mögliche Werte:** Beliebige Zeichenkette mit dem Namen der Region, z. B. *us-east-2* Buckets, die in einer bestimmten Region erstellt wurden, können nicht von einer anderen Region aus aufgerufen werden, siehe Regionen und Endpoints.

`bucket`	Bucket-Name Der Bucket-Name muss innerhalb von Amazon S3 eindeutig sein, siehe Buckets. **Mögliche Werte:** Beliebige Zeichenkette mit dem Bucket-Namen Folgende Regeln gelten für die Benennung von S3-Buckets in allen AWS-Regionen. Bucket-Namen ... • müssen innerhalb von Amazon S3 eindeutig sein • müssen den DNS-Namenskonventionen entsprechen • müssen mindestens 3 und höchstens 63 Zeichen lang sein • dürfen keine Großbuchstaben oder Unterstriche enthalten • müssen mit einem Kleinbuchstaben oder einer Zahl beginnen • müssen eine Reihe aus einem oder mehreren Zeichen sein Aufeinander folgende Namen werden durch einen einzelnen Punkt (`.`) getrennt. Bucket-Namen können Kleinbuchstaben, Zahlen und Bindestriche enthalten. Jeder Name muss mit einem Kleinbuchstaben oder einer Zahl beginnen und enden. • dürfen nicht als IP-Adresse formatiert sein, z. B. `192.168.5.4` • Beim Verwenden von virtuellen gehosteten Buckets mit SSL (Secure Sockets Layer) trifft das SSL-Wildcard-Zertifikat nur auf Buckets zu, die keine Punkte enthalten. Um dies zu umgehen, verwenden Sie HTTP oder schreiben Sie Ihre eigene Zertifikatsprüfungs-Logik. Bei virtuellen gehosteten Buckets empfehlen wir keine Punkte (`.`) im Bucket-Namen zu verwenden.
`bucketVersion`	Bucket-Versionierung, um mehrere Varianten eines Objekts im selben Bucket zu halten **Mögliche Werte:** • `OFF`: Die Versionierung ist deaktiviert (Standard) Ist die Versionierung für ein Bucket aktiviert, kann sie nicht mehr deaktiviert und auf den Status `OFF` gesetzt werden. Die Versionierung kann lediglich ausgesetzt und auf den Status `SUSPENDED` gesetzt werden. • `ENABLED`: Die Bucket-Versionierung ist aktiviert • `SUSPENDED`: Die Bucket-Versionierung wird ausgesetzt, wenn die Versionierung vorher aktiviert war
`publicAccess`	Gibt an, ob Buckets oder Dateien bei der Operation `AddGrantsPermission` öffentlich zugänglich sind oder nicht

<table>
<tr><td><code>publicPermission</code></td><td>Konfiguration für den öffentlichen Zugang

Mögliche Werte:

<table>
<tr><th>Berechtigung</th><th>Bei einem Bucket gestattet</th><th>Bei einem Objekt gestattet</th></tr>
<tr><td><code>READ</code></td><td>Erlaubt es, die Objekte im Bucket aufzulisten</td><td>Erlaubt es, die Objekt-Daten und -Metadaten zu lesen</td></tr>
<tr><td><code>WRITE</code></td><td>Erlaubt es, ein beliebiges Objekt im Bucket zu erstellen, zu überschreiben und zu löschen</td><td>Nicht anwendbar</td></tr>
<tr><td><code>READ_ACP</code></td><td>Erlaubt es, die Bucket-ACP zu lesen</td><td>Erlaubt es, die Objekt-ACP zu lesen</td></tr>
<tr><td><code>WRITE_ACP</code></td><td>Erlaubt es, die ACP für den jeweiligen Bucket zu schreiben</td><td>Erlaubt es, die ACP für das jeweilige Objekt zu schreiben</td></tr>
<tr><td><code>FULL_CONTROL</code></td><td>Erlaubt Zugriff auf die Bucket-Berechtigungen <code>READ</code>, <code>WRITE</code>, <code>READ_ACP</code> und <code>WRITE_ACP</code></td><td>Erlaubt Zugriff auf die Objekt-Berechtigungen <code>READ</code>, <code>WRITE</code>, <code>READ_ACP</code> und <code>WRITE_ACP</code></td></tr>
</table>
</td></tr>
<tr><td><code>file</code></td><td>Name der zu speichernden Datei, siehe Objekte

Folgende Zeichensätze sind im Allgemeinen in Schlüssel-Namen zulässig:

• Alphanumerische Zeichen <code>[0-9a-zA-Z]</code>
• Sonderzeichen: <code>!</code>, <code>-</code>, <code>_</code>, <code>.</code>, <code>*</code>, <code>'</code>, <code>(</code>, und <code>)</code>

Mögliche Werte: Beliebige Zeichenkette mit dem Objekt-Schlüssel-Namen, z. B.

• <code>4my-organization</code>
• <code>my.great_photos-2014/jan/myvacation.jpg</code>
• <code>videos/2014/birthday/video1.wmv</code></td></tr>
<tr><td><code>path</code></td><td>Pfad der hochzuladenden Datei

Mögliche Werte: Beliebiger gültiger Pfad, z. B.

• X4DB-Pfad: <code>x4db://1/.....</code>
• XSTORE-Pfad <code>xstore://</code>
• Vollständiger Pfad: <code>C:/Documents/....</code></td></tr>
<tr><td><code>overwrite</code></td><td>Information, ob die Datei überschrieben werden soll oder nicht

Mögliche Werte:

• <code>true</code>: Überschreibt die Datei
• <code>false</code>: Überschreibt die Datei nicht</td></tr>
</table>

`targetBucket`	Name des Buckets, in dem die Datei kopiert oder verschoben werden soll **Mögliche Werte:** Siehe Parameter `bucket`
`targetFile`	Name der Datei, wenn sie unter dem neuen Pfad kopiert oder verschoben wird **Mögliche Werte:** Siehe Parameter `file`
`includeObjectsMetadata`	Gibt an, ob Objekt-Metadaten als Teil der Antwort einbezogen werden oder nicht **Possible values:** • `true`: Objekt-Metadaten werden als Teil der Antwort einbezogen • `false`: Objekt-Metadaten werden nicht als Teil der Antwort einbezogen

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt
`0`	Die Ausführung der Operation wurde noch nicht abgeschlossen oder es wurde kein Ergebnis ausgegeben
`-1`	Bei der Ausführung des Adapters kam es zu einem Fehler

Input

Der Adapter erwartet ein beliebiges Input-Dokument, z. B. ein Binär- oder ein Base-64-Dokument (XML oder Text).

- Operation `Upload`: Ist der Parameter `path` angegeben, wird kein Input-Dokument vorausgesetzt. In allen anderen Fällen wird ein Input-Dokument erwartet.
- Operation `Download` und `Search`: Es wird kein Input-Dokument vorausgesetzt.
- Operation `AddGrantsPermission`:

Beispiel-Input für die Operation AddGrantsPermission

```
<?xml version='1.0' encoding='UTF-8'?>
<RootElement>
    <Grant id="Canonical User ID">
        <permission> Read </permission>
        <permission> Write </permission>
        <permission> WriteAcp </permission>
        <permission> ReadAcp </permission>
        <permission> FullControl </permission>
    </Grant>
</RootElement>
```

Um Objekt-Berechtigungen hinzuzufügen, sind die Parameter `bucket` und `file` erforderlich. Um Bucket-Berechtigungen hinzuzufügen, ist der Parameter `bucket` erforderlich und `file` muss leer sein.

- Operation `RevokeGrantsPermission`:

Beispiel-Input für die Operation RevokeGrantsPermission

```
<?xml version='1.0' encoding='UTF-8'?>
<RootElement>
    <Grant id="Canonical User ID">
    </Grant>
</RootElement>
```

Um Objekt-Berechtigungen hinzuzufügen, sind die Parameter `bucket` und `file` erforderlich. Um Bucket-Berechtigungen hinzuzufügen, ist der Parameter `bucket` erforderlich und `file` mu ss leer sein.

- Für die Operation `AddBucketPolicy` muss der Input im JSON-Format erfolgen
  Unter folgenden Links finden Sie Zusatzinformationen, wie Beispiele für Bucket-Richtlinien, Berechtigungen in einer Richtlinien und Bedingungen in Richtlinien.
- Operation `DeleteMultiFiles`:

Beispiel-Input für die Operation DeleteMultiFiles

```
<?xml version='1.0' encoding='UTF-8'?>
<RootElement>
    <Object name="File Name"></Object>
    <Object name="File Name"></Object>
    <Object name="File Name"></Object>
    <Object name="File Name"></Object>
</RootElement>
```

Ausgabe

Der Adapter gibt abhängig von der ausgeführten Operation verschiedene Dokumente aus:

- Operation `Upload`:
  - Wurde der Adapter erfolgreich ausgeführt (Status `1`), wird ein XML-Dokument ausgegeben.
  - Kam es zu einem Fehler während der Adapter-Ausführung (Status `-1`), wird nichts ausgegeben.
- Operation `Search`:
  - Wurde der Adapter erfolgreich ausgeführt (Status `1`), wird ein XML-Dokument mit den verwendeten Filtern ausgegeben.

Beispiel-Output für die Operation SEARCH

```
<?xml version='1.0' encoding='UTF-8'?>
<ListBucket>
    <Bucket name="test-malaga2" region="eu-west-1">
        <Object name="myFile.xml" url="https://s3-eu-west-1.amazonaws.com/
test-malaga2/myFile.xml"/>
        <Object name="myFile2.xml" url="https://s3-eu-
west-1.amazonaws.com/test-malaga2/myFile2.xml"/>
        <Object name="pic2.jpg" url="https://s3-eu-west-1.amazonaws.com/
test-malaga2/pic2.jpg"/>
        <Object name="testing_folders/" url="https://s3-eu-
west-1.amazonaws.com/test-malaga2/testing_folders/"/>
        <Object name="testing_folders/personalData.txt" url="https://s3-
eu-west-1.amazonaws.com/test-malaga2/testing_folders/personalData.txt"/>
    </Bucket>
</ListBucket>
```

- Wurde der Adapter erfolgreich ausgeführt (Status `1`) und der Parameter `IncludeObjectsMetada` ist ausgewählt , werden zusätzlich die Metadaten für jedes Objekt ausgegeben:

Beispiel-Output für die Operation SEARCH inklusive Metadaten

```
<?xml version='1.0' encoding='UTF-8'?>
<ListBucket>
    <Bucket name="test-malaga2" region="eu-west-1">
        <Object name="myFile.xml" url="https://s3-eu-west-1.amazonaws.com/
test-malaga2/myFile.xml">
            <Metadata lastModified="2018-12-10 13:42:38.000 +0100"
instanceLength="158" eTag="a933d9ca99b473c7921c496727a32f91" contentType="
application/octet-stream" contentLength="158"/>
        </Object>
        <Object name="myFile2.xml" url="https://s3-eu-
west-1.amazonaws.com/test-malaga2/myFile2.xml">
            <Metadata lastModified="2018-12-10 12:02:07.000 +0100"
instanceLength="158" eTag="a933d9ca99b473c7921c496727a32f91" contentType="
application/octet-stream" contentLength="158" contentLanguage="English"/>
        </Object>
        <Object name="pic2.jpg" url="https://s3-eu-west-1.amazonaws.com/
test-malaga2/pic2.jpg">
            <Metadata lastModified="2018-12-10 11:37:52.000 +0100"
instanceLength="59580" eTag="fb913694d969758b431277b366f53d7" contentType="
image/jpeg" contentLength="59580" contentLanguage="Spanish"/>
        </Object>
        <Object name="testing_folders/" url="https://s3-eu-
west-1.amazonaws.com/test-malaga2/testing_folders/">
            <Metadata lastModified="2018-12-10 12:24:54.000 +0100" eTag="d
41d8cd98f00b204e9800998ecf8427e" contentType="application/x-directory"/>
        </Object>
        <Object name="testing_folders/personalData.txt" url="https://s3-
eu-west-1.amazonaws.com/test-malaga2/testing_folders/personalData.txt">
            <Metadata lastModified="2018-12-10 12:30:35.000 +0100"
instanceLength="33" eTag="7960fb01e91834eb9d5ccbacdf45d0e5" contentType="a
pplication/octet-stream" contentLength="33" contentLanguage="Spanish"/>
        </Object>
    </Bucket>
</ListBucket>
```

    - Kam es zu einem Fehler während der Adapter-Ausführung (Status `-1`), wird nichts ausgegeben.
- Operation `Download`:
    - Wurde der Adapter erfolgreich ausgeführt (Status `1`), wird der heruntergeladene Blob (Text, XML, Binärdokument etc.) ausgegeben.
    - Existieren das Objekt oder der Bucket nicht, wird der Status `-1` zurückgegeben.
- Operation `GetAccessControlList`:

Beispiel-Output für die Operation GetAccessControlList

```
<AccessControlList>
    <Object Grant="e0188cd1cbfbb71222450e8e416e138172b036cdf3646ab51fb0ae6334d5
ae51" Permission="FULL_CONTROL"/>
    <Object Grant="7779a9de4e188ca2c3fd5ef47f12f2a0fd3b9b30d283e343e0a55746d8df
4802" Permission="READ"/>
    <Object Grant="7779a9de4e188ca2c3fd5ef47f12f2a0fd3b9b30d283e343e0a55746d8df
4802" Permission="READ_ACP"/>
</AccessControlList>
```

- Operation `GetBucketPolicy` :

```
{
"Version":"2012-10-17", "Statement":[
}
"Sid":"AddPerm",
"Effect":"Allow",
"Principal":"*",
"Action":"s3:GetObject",
"Resource":"arn:aws:s3:::softprojectamazons3input/*"
}
]
}
```

10.3.2 Amazon SimpleDB Connector

`Amazon SimpleDB Connector`: Amazon SimpleDB ist ein hochverfügbarer NoSQL-Datenspeicher, der die Arbeit der Datenbankverwaltung entlastet.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `CreateDomain`: Legt eine Domain mit Datensätzen an. **Erforderliche Parameter:** `domain` • `DomainMetadata`: Ruft Metadaten zu einer Domain auf, siehe DomainMetadata im offiziellen Handbuch. **Erforderliche Parameter:** `domain` • `DeleteDomain`: Löscht eine Domain. **Erforderliche Parameter:** `domain` • `ListDomain`: Listet alle Domains auf. • `Put`: Fügt ein Element und seine Attribute hinzu oder aktualisiert sie, oder fügt Attribut-Wert-Paare zu einem bereits vorhandenen Element hinzu; Das Element wird beim Empfang automatisch indiziert. Gemäß dem Parameter `replaceExistingAttributes` ersetzt der Inhalt bestehende Attribute oder erstellt neue Attribute, wobei auch die aktuellen Attribute erhalten bleiben. **Erforderliche Parameter:** `domain`, `replaceExistingAttributes` • `BatchPutAttributes`: Fügt bis zu 25 `Put`-Operationen in einem einzigen Anruf hinzu oder aktualisiert sie. Elemente werden beim Empfang automatisch indiziert. Gemäß dem Parameter `replaceExistingAttributes` ersetzt der Inhalt bestehende Attribute oder erstellt neue Attribute, wobei auch die aktuellen Attribute erhalten bleiben, siehe BatchPutAttributes im offiziellen Handbuch. **Erforderliche Parameter:** `domain`, `replaceExistingAttributes` • `GetItem`: Ruft ein Element und alle oder eine Teilmenge seiner Attribute und Werte ab. **Erforderliche Parameter:** `domain`, `itemName` • `DeleteItem`: Löscht ein Element, ein Attribut oder einen Attribut-Wert. **Erforderliche Parameter:** `domain`, `itemName` • `BatchDeleteAttributes`: Löscht bis zu 25 `Delete`-Operationen in einem einzigen Anruf, siehe BatchDeleteAttributes im offiziellen Handbuch. **Erforderliche Parameter:** `domain` • `Select`: Fragt einen Datensatz gemäß der typischen Abfragesyntax *“`select target from domain_name where query_expression`”* ab. Unterstützte Wertetests sind: `=`, `!=`, `like`, `not like`, `between`, `is null`, `is not null`, und `every ()`. *Beispiel:* *`select * from mydomain where every(keyword) = ‘Book’`*. Ordnen Sie Ergebnisse mit dem Operator `SORT` und zählen Sie Elemente, welche die Abfragebedingung(en) erfüllen, mit dem Operator `Count`. • `StoreAmazonS3ObjectMetadata`: Speichert die Metadaten zu einem AmazonS3-Objekt in der im Parameter `domain` spezifizierten Domain. **Erforderliche Parameter:** `domain`, `replaceExistingAttributes`

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.amazonsimpledb.AmazonSimpleDBAdapter*: Hauptklasse (Standard)
`keyId`	Zugangsschlüssel-ID für Anfragen an AWS (Amazon Web Services) Die Zugangsdaten werden über Parameter an den Adapter gesendet. Dabei gibt es verschiedene Konfigurationsmöglichkeiten, siehe Zugangsdaten einrichten. **Mögliche Werte:** Beliebige Zeichenkette mit der Zugangsschlüssel-ID, z. B. *`AKIAIOSFODNN7EXAMPLE`*
`secretKey`	Geheimer Zugangsschlüssel für Anfragen an AWS Die Zugangsdaten werden über Parameter an den Adapter gesendet. Dabei gibt es verschiedene Konfigurationsmöglichkeiten, siehe Zugangsdaten einrichten. **Mögliche Werte:** Beliebige Zeichenkette mit der geheimen Zugangsschlüssel, z. B. *`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`*
`region`	Region der Domains **Mögliche Werte:** Beliebige Zeichenkette mit dem Namen der Region, z. B. *`us-east-2`* Buckets, die in einer bestimmten Region erstellt wurden, können nicht von einer anderen Region aus aufgerufen werden, siehe Regionen und Endpoints.
`domain`	Domain-Name, der durch einen eindeutigen Schlüssel (Namen) identifiziert wird. Domains werden durch Domain-Registerkarten am unteren Rand der Tabelle dargestellt und ähneln Tabellen mit ähnlichen Daten. Der Name muss zwischen 3 und 255 Zeichen lang sein und kann folgende Zeichen enthalten: `a-z`, `A-Z`, `0-9`, '`_`', '`-`' und '`.`'. Es können Abfragen gegen eine Domain, jedoch nicht zwischen verschiedenen Domains ausgeführt werden.
`itemName`	Element-Name Elemente werden durch Zeilen dargestellt und sind Einzelobjekte, die ein oder mehrere Attribut-Namens-Wertepaare enthalten.
`replaceExistingAttributes`	Gibt an, ob der zu setzende Inhalt den bestehenden überschreibt, wenn das zu setzende Element) ein gemeinsames Attribut enthält. Gilt für die Operationen `Put` und `BatchPutAttributes`.

Status

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt
`0`	Die Domain existiert nicht
`-1`	Bei der Ausführung des Adapters kam es zu einem Fehler

Input

Der Adapter erwartet ein XML-Dokument als Input

- Operation `Put`:

```
<?xml version="1.0" encoding="UTF-8"
<ListAttributes>
    <item name="..Item name..">
        <Attribute name="..Name of Attribute.." value="..Value of Attribute.."/
>
        <!-- As many attributes as required -->
    </item>
</ListAttributes>
```

> Bei der Operation `Put` sollte nur ein Element-Attribute angegeben werden. Wenn Operationen für mehr als ein Element ausgeführt werden sollen, die Operation `BatchPutAttributes` verwenden. Es gibt keine Einschränkungen bei den Elementnamen, allerdings ist das Attribut `name` erforderlich.

- Operation `BatchPutAttributes` :

```
<?xml version="1.0" encoding="UTF-8"
<ListAttributes>
    <item name="..Item name..">
        <Attribute name="..Name of attribute.." value="..Value of attribute.."/
>
        <!-- As many attributes as required -->
    </item>
    <!-- As many items as required -->
</ListAttributes>
```

> Bei der Operation `BatchPutAttributes` sollten mehrere Element-Attribute angegeben werden. Wenn Operationen für ein einziges Element ausgeführt werden sollen, die Operation `Put` verwenden. Es gibt keine Einschränkungen bei den Elementnamen, allerdings ist das Attribut `name` erforderlich.

- Operation `Select`:

```
<?xml version="1.0" encoding="UTF-8"
<RootElement>
    <queryStatement>
        select * from 'DomainName'
    </queryStatement>
</RootElement>
```

> Das Element `queryStatement` wird benötigt. Für eine korrekte Abfrage, folgende Anleitung befolgen.

- Operation `BatchDeleteAttributes`:

```
<?xml version="1.0" encoding="UTF-8"
<Items>
    <item name="HR employee">
        <Attribute1 name="ID" value="00002"/>
    </item>
    <item3 name="DEV employee">
        <Attribute1 name="ID" value="00003"/>
        <Attribute2 name="Address" value="C\Ettlingen"/>
        <Attribute3 name="Name" value="Lorena"/>
        <Attribute4 name="Frameworks" value="Java-Hibernate"/>
        <Attribute4 name="Frameworks" value="Angular"/>
    </item>
</Items>
```

- Operation `StoreAmazonS3ObjectMetadata`:

```
<?xml version='1.0' encoding='UTF-8'?>
<ListBucket>
    <Bucket name="SoftProject-bucket" region="eu-west-1">
        <Object name="myFile.xml" url="https://s3-eu-west-1.amazonaws.com/test-
malaga2/myFile.xml">
            <Metadata lastModified="2018-12-10 13:42:38.000 +0100"
instanceLength="158" eTag="a933d9ca99b473c7921c496727a32f91" contentType="appli
cation/octet-stream" contentLength="158"/>
        </Object>
        <Object name="myFile2.xml" url="https://s3-eu-west-1.amazonaws.com/
test-malaga2/myFile2.xml">
            <Metadata lastModified="2018-12-10 12:02:07.000 +0100"
instanceLength="158" eTag="a933d9ca99b473c7921c496727a32f91" contentType="appli
cation/octet-stream" contentLength="158" contentLanguage="English"/>
        </Object>
    </Bucket>
</ListBucket>
```

- Die erwartete Input-XML-Struktur kann mit Hilfe der Operation `Search` des Amazon S3 Simple Storage Connectors erzeugt werden. Dabei muss die Adapter-Eigenschaft `includeObjectsMetadata` aktiviert sein.
- Stored attributes for each object metadata need to match with the Amazon metadata fields, see Object Key and Metadata.

Additionally to the provided metadata fields, a new field will be stored in order to be able to filter the metadata. The field `metadataType` provides the values `STANDARD_METADATA` or `USER_METADATA` or both, if there are standard fields and user fields, see Object metadata.

```
<?xmlversion='1.0'encoding='UTF-8'?>
<ListBucket>
    <Bucketname="testmalaga2"region="euwest1">
        <Objectname="testing_folders/personalData.txt"url="https://
s3euwest1.amazonaws.com/testmalaga2/testing_folders/personalData.txt">
            <MetadatalastModified="2018-12-10 12:30:35.000 +0100"instanceLength="
33"eTag="7960fb01e91834eb9d5ccbacdf45d0e5"contentType="application/octetstream"c
ontentLength="33"contentLanguage="Spanish"/>
            <UserMetadataciudad="Malaga"/>
        </Object>
    </Bucket>
</ListBucket>
```

Ausgabe

Der Adapter gibt je nach ausgeführter Operation unterschiedliche Arten von Dokumenten zurück:

- Operation `ListDomain`:

```
<?xml version="1.0" encoding="UTF-8"
<ListDomain>
    <Domain name="Domain Name">
</ListDomain>
```

- Operation `GetItem`:

```
<?xml version="1.0" encoding="UTF-8"
<Result>
    <Item id="Item Name">
        <Attribute name="Attribute name" value="Attribute value"/>
        <Attribute name="Attribute name" value="Attribute value"/>
        <Attribute name="Attribute name" value="Attribute value"/>
    </Item>
</Result>
```

- Operation `Select`:

```
<?xml version="1.0" encoding="UTF-8"
<Result>
    <Item id="Item1 Name">
        <Attribute name="Attribute name" value="Attribute value"/>
        <Attribute name="Attribute name" value="Attribute value"/>
        <Attribute name="Attribute name" value="Attribute value"/>
    </Item>
    <Item id="Item2 Name">
        <Attribute name="Attribute name" value="Attribute value"/>
        <Attribute name="Attribute name" value="Attribute value"/>
        <Attribute name="Attribute name" value="Attribute value"/>
    </Item>
    <Item id="Item3 Name">
        <Attribute name="Attribute name" value="Attribute value"/>
        <Attribute name="Attribute name" value="Attribute value"/>
        <Attribute name="Attribute name" value="Attribute value"/>
    </Item>
</Result>
```

- Alle anderen Operationen geben das Input-Dokument zurück.

10.3.3 Dropbox Connector

`Dropbox Connector`: Stellt eine Verbindung zum Dropbox-Cloud-Dienst her und bietet Funktionen zur Verwaltung von Cloud-Speicherdaten.

> ⚠ Für die Verwendung des Dropbox Connectors wird ein Dropbox-Konto (Standard oder Business) benötigt.

Vor der ersten Verwendung des Adapters:

Wenn der Adapter Dropbox Connector zum ersten Mal verwendet werden soll, dann müssen zuvor folgende Schritte ausgeführt werden:

1. Die Website https://www.dropbox.com/developers aufrufen.
2. Auf **App Console** klicken. Der Button befindet sich oben rechts.
3. Auf **Create app** klicken.
    a. Für **Choose an API** `Dropbox API` auswählen.
    b. Für **Choose the type of access you need** `Full Dropbox` auswählen.
    c. Für **Name your app** einen beliebigen Namen eingeben.
    d. Auf **Create app** klicken
    e. Die App wird erzeugt und die App-Einstellungen öffnen sich.
4. Unter **Generated access token** auf **Generate** klicken.
5. Der Access Token wird erzeugt und angezeigt. Dieser Access Token wird für die Verwendung des Dropbox Connectors benötigt.

Über die API wird in der zugeordneten Dropbox ein Ordner `Apps` erstellt, dieser bildet den Anwendungsordner. Jeder, der im Besitz des generierten Access Token ist, hat Zugriff auf diesen Ordner! Achten Sie darauf, wer in den Besitz des Access Token gelangt!

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `CopyFile`: Kopiert eine Datei innerhalb der Dropbox. • `CreateFile`: Fügt der Dropbox eine Datei hinzu. • `DeleteFile`: Löscht eine Datei innerhalb der Dropbox. • `ExtractFolder`: Extrahiert eine .zip-Datei in den bei den Parametern angegebenen Ordner. • `GetFileContent`: Gibt den Inhalt einer Datei aus. • `GetFileMetadata`: Gibt die Metadaten einer Datei aus. • `ListFolder`: Listen den Inhalt eines Ordners auf. • `UpdateFile`: Aktualisiert eine Datei.

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.adapter.dropbox.DropBoxConnector*: Hauptklasse (Standard)
`accessToken`	*Erforderlich*. Wird benötigt, um eine Verbindung mit dem Dropbox-Konto herzustellen. **Mögliche Werte:** eindeutiger Access Token Der Access Token muss in der Dropbox generiert werden.

`folderPath`	Pfad zu einem Ordner in der Dropbox. **Mögliche Werte:** Ordnerpfad (Standard: `/` (oberste Ordnerebene)) ⚠ Der Ordnerpfad muss immer mit einem `/` beginnen und enden. ⓘ Wenn während der Operation `CreateFile` Ordner verwendet werden, die nicht existieren, dann werden diese automatisch erstellt.
`fileName`	Dateiname der Datei, die verarbeitet werden soll. **Mögliche Werte:** Dateiname inkl. Dateiendung
`sourcePath`	*Erforderlich für die Operation `CopyFile`.* Gibt an, welche Datei kopiert werden soll. **Mögliche Werte:** Dateipfad inkl. Dateiname und Dateiendung, z. B. `folder/file.txt`
`destinationPath`	*Erforderlich für die Operation `CopyFile`.* Gibt an, welche Datei kopiert werden soll. **Mögliche Werte:** Dateipfad
`folderName`	*Erforderlich für die Operation `ExtractFolder`.* Name des Ordners, in dem die extrahierten Dateien erstellt oder überschrieben werden. **Mögliche Werte:** Beliebige Zeichenkette
`overwrite`	Definiert, ob vorhandene Dateien beim Kopieren oder Erstellen überschrieben werden. **Mögliche Werte:** • `true`: Wenn eine Datei mit dem gleichen Namen an diesem Speicherort bereits existiert, dann wird diese Datei überschrieben. • `false`: Wenn eine Datei mit dem gleichen Namen an diesem Speicherort bereits existiert, dann wird die neue Datei umbenannt, z. B. von `File.txt` zu `Copy_of_File.txt`.

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.

`-1`	Die Operation konnte nicht ausgeführt werden. Die Fehlermeldung wird als Textdokument und im Server-Log ausgegeben.

Input

Der Adapter erwartet für die Operationen `CreateFile` und `UpdateFile` eine Datei als Input.

Der Adapter erwartet für die Operation `ExtractFolder` eine .zip-Datei als Input.

Output:

Der Adapter liefert abhängig von der ausgeführten Operation unterschiedliche Daten als Output.

- Operationen `CopyFile`, `CreateFile` und `DeleteFile`: Textdokument mit Information
- Operationen `ListFolder`, `GetFileMetadata`: XML-Dokument mit entsprechendem Inhalt

Beispieloutput GetFileMetadata

```
<Metadata isFolder="false">
    <Name>new.txt</Name>
    <Path>/new.txt</Path>
    <Id>id:SLz0S6oc-FAAAAAAAAALw</Id>
    <LastModified>2019-06-26T11:55:33</LastModified>
    <Size>4</Size>
</Metadata>
```

Beispieloutput ListFolder

```
<Result>
    <Entry isFolder="true">Folder</Entry>
    <Entry isFolder="false">Image.png</Entry>
    <Entry isFolder="false">file.txt</Entry>
    <Entry isFolder="false">Document.docx</Entry>
</Result>
```

- Operation `ExtractFolder`: Liefert keinen Output.
- Operation `GetFileContent`: Inhalt der ausgelesenen Datei

⚠ Es werden nicht alle Dateiformate unterstützt!

10.3.4 Google Cloud Storage Objects Connector

`Google Cloud Storage Objects Connector`: Ermöglicht das Speichern, Suchen, Herunterladen und das Löschen von Objekten in der Google Cloud.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation
	
	Mögliche Werte:
	
	• `Search`: Sucht nach Informationen (Buckets oder Objekte) im Google-Cloud-Speicher
	Erforderliche Parameter: `bucketName`, `objectName`
	
	• `Upload`: Lädt Buckets oder Objekte in den Google-Cloud-Speicher hoch oder legt Buckets oder Objekte im Google-Cloud-Speicher an
	Erforderliche Parameter: `bucketName`, `objectName`, `bucketRegion`, `storageClass`, `fileLocation`, `contentType`
	
	• `Download`: Lädt ein Objekt vom Google-Cloud-Speicher herunter
	Erforderliche Parameter: `bucketName`, `objectName`
	
	• `Delete`: Löscht ein Bucket oder Objekt im Google-Cloud-Speicher
	Erforderliche Parameter: `bucketName`, `objectName`
	
	• `CopyObject`: Kopiert ein Objekt von einem Bucket zu einem anderen. Es gibt mehrere Möglichkeiten, ein Objekt zu kopieren:
	Erforderliche Parameter: `bucketName`, `objectName`
	
	  • Objekt innerhalb eines Buckets kopieren und den Namen ändern.
	  Erforderliche Parameter: `targetObjectName`
	
	  • Objekt in ein anderes Bucket kopieren und den Namen ändern.
	  Erforderliche Parameter: `targetObjectName`, `targetBucketName`
	
	  • Objekt in ein anderes Bucket kopieren und den Namen beibehalten.
	  Erforderliche Parameter: `targetBucketName`
	
	• `MoveOrRenameBucket`: Verschiebt ein komplettes Bucket oder benennt es um.
	Erforderliche Parameter: `bucketName`, `targetBucketName`, `targetBucketRegion`
	
	ⓘ • Wenn der Name des Ziel-Buckets bereits existiert, dann wird eine Fehlermeldung ausgegeben.
	• Wenn eines der Objekte im Ziel-Bucket bereits existiert und die `overwrite`-Option gewählt ist, dann werden die vorhandenen Objekte überschrieben. Wenn die `overwrite`-Option nicht gewählt ist, dann wird eine Fehlermeldung ausgegeben.

> Da ein Bucket zu einer anderen geografischen Region verschoben werden kann, muss auf die Beschränkungen in Verbindung mit dem Verschieben in andere geografische Regionen geachtet werden. Weitere Informationen zu Bucket-Standorten.

- `MoveOrRenameObject`: Verschiebt ein Objekt oder benennt es um.
  Erforderliche Parameter: `bucketName`, `objectName`, `targetBucketName`, `targetObjectName`

  > Wenn das Objekt im Zielbucket bereits existiert und die `overwrite`-Option gewählt ist, dann wird das vorhandene Objekte überschrieben. Wenn die `overwrite`-Option nicht gewählt ist, dann wird eine Fehlermeldung ausgegeben.

- `UpdateObjectStorageClass`: Ändert die zum Objekt zugeordnete `storageClass`. Die Eigenschaft `storageClass` ist eng mit der Bucket-Region verbunden. Beim Anlegen eines Buckets muss beispielsweise die Region angegeben werden.
  Erforderliche Parameter: `bucketName`, `objectName`, `storageClass`

  > Nicht alle möglichen Werte sind für jedes Objekt/Bucket erlaubt. Es hängt von der Bucket-Spezifikation und dem Objekt selbst ab. Weitere Informationen zu Storage-Klassen.

  > Ein regionales Bucket kann nur regionale Objekte speichern. In diesem Fall kann die `storageClass` eines Objektes nur in `nearline` oder `coldline` geändert werden.
  >
  > Ein multi-regionales Bucket kann nur multi-regionale Objekte speichern. In diesem Fall kann die `storageClass` eines Objektes nur in `nearline` oder `coldline` geändert werden.
  >
  > **Mögliche Werte:**
  >
  > - Objekt ist in einem regionalen Bucket oder einem multi-regional Bucket gespeichert: `coldline`, `nearline`
  > - Objekt ist in einem nearline-Bucket gespeichert, Bucket-Region ist regional oder multi-regional: `coldline`, `regional` Wenn die Bucket-Region nicht regional oder multi-regional ist: nur `coldline`
  > - Objekt ist in einem coldline-Bucket gespeichert, Bucket-Region ist regional oder multi-regional: `nearline`, `regional`. Wenn die Bucket-Region nicht regional oder multi-regional ist: nur `nearline`

> Wenn eine regionale Region spezifiziert wird, z. B. `europe-west1`, wird implizit festgelegt, dass die Default-`storageClass` regional, nearline oder coldline ist, aber nie multi-regional.
>
> Das selbe passiert, wenn eine Region als multiregional, z. B. `us`, spezifiziert wird: Es wird implizit festgelegt, dass die Default-`storageClass` multi-regional, nearline oder coldline ist, aber nie regional.

- `AddGrantsPermission`: Fügt eine ACL-Erlaubnis zu einem Objekt oder Bucket hinzu. Wenn das Objekt oder Bucket bereits eine ACL-Erlaubnis hat, passiert nichts.

  > **Erforderliche Parmeter:**
  >
  > - Für diesen Vorgang sind die Parameter `permissionEntity`, `permissionRole`, `permissionName` und falls vergeben `permissionProjectRole` erforderlich.
  > - Erlaubnis einem Objekt hinzufügen: `objectName` und `bucketName`
  > - Erlaubnis einem Bucket hinzufügen: `bucketName`

  > Wenn eine neue Erlaubnis zu einem Bucket hinzugefügt wird, erhält das Bucket die Erlaubnis, aber nicht die enthaltenen Objekte. Das bedeutet, dass bei einem öffentlichen Bucket jeder das Bucket und die Liste der enthaltenen Objekte sehen kann, aber nicht auf die Objekte zugegriffen werden kann. Dafür wird eine zusätzliche Erlaubnis benötigt.

- `RevokeGrantsPermission`: Entfernt eine ACL-Erlaubnis von einem Objekt oder Bucket. Wenn das Objekt oder Bucket keine ACL-Erlaubnis hat, passiert nichts.

  > **Erforderliche Parmeter:**
  >
  > - Für diesen Vorgang sind die Parameter `permissionEntity`, `permissionRole`, `permissionName` und falls vergeben `permissionProjectRole` erforderlich.
  > - Erlaubnis von einem Objekt entfernen: `objectName` und `bucketName`
  > - Erlaubnis von einem Bucket entfernen: `bucketName`

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.google.cloud.storage.GoogleCloudStorageObjectAdapter*: Hauptklasse (Standard)
`serviceAccountFile`	Pfad zur JSON-Datei mit dem Service-Account-Schlüssel zur Authentifizierung gegenüber dem Google-Cloud-Speicher. **Mögliche Werte:** • System-Pfad, z. B. *`C:\X4\V_5.9\Server\X4DB\1\ExampleProcess\GoogleCloud\Config\serviceAccountFile.json`* • X4db-URL, z. B. *`x4db://1/ExampleProcess/GoogleCloud/Config/serviceAccountFile.json`* • Xstore-URL, z. B. *`xstore://ExampleProcess/GoogleCloud/Config/serviceAccountFile.json`*
`bucketName`	Name des Buckets, in dem die Objekte organisiert sind **Mögliche Werte:** Zeichenkette mit dem Namen Bucket-Namen müssen den folgenden Namenskonventionen entsprechen: • Bucket-Namen dürfen nur Kleinbuchstaben, Zahlen, Bindestriche (`-`), Unterstriche (`_`) und Punkte (`.`) enthalten. Namen, die Punkte enthalten, müssen überprüft werden. • Bucket-Namen müssen mit einer Zahl oder einem Buchstaben beginnen und enden. • Bucket-Namen müssen zwischen 3 und 63 Zeichen enthalten. Namen, die Punkte enthalten, können aus bis zu 222 Zeichen bestehen, jedoch darf jeder durch einen Punkt getrennte Teil nicht länger als 63 Zeichen sein. • Bucket-Namen können nicht aus IP-Adressen mit dem Beispiel-Format *192.168.5.4* bestehen • Bucket-Namen dürfen nicht mit dem Präfix `goog` beginnen. • Bucket-Namen dürfen weder `google` noch ähnliche Schreibweisen für Google wie beispielsweise `g00gle` enthalten.
`targetBucketName`	*Optional*. Name des Ziel-Buckets. Für Operationen, die ein Bucket kopieren/verschieben/umbenennen. Der Name unterliegt denselben Beschränkungen wie der Parameter `bucketName`.

`bucketRegion`	Region, in der die Buckets definiert sind **Mögliche Werte:** Zeichenkette mit der Region *Beispiel*: • Regionaler Standort: `europe-west3` (Europa, Frankfurt) • Multi-regionaler Standort: `eu` (Rechenzentren in der Europäischen Union)
`targetBucketRegion`	Name der Region des Ziel-Buckets. Für Operationen, die ein Bucket verschieben/umbenennen und für die eine Zielregion angegeben werden kann. Der Parameter unterliegt denselben Beschränkungen wie der Parameter `bucketRegion`.
`storageClass`	Speicher-Klasse des Buckets und der Objekte. Dieser Parameter ist eng verbunden mit Regionen, da ein `storageClass`-Wert nur einem Bucket oder einem Objekt zugeordnet werden kann, das mit dem gewählten `bucketRegion`-Wert verbunden ist. **Mögliche Werte:** Auswahl mit folgenden Werten • `multi_regional` • `regional` • `nearline` • `coldline`
`objectName`	Name des Objekts. Erforderlich für die Operationen `AddGrantsPermission` und `RevokeGrantsPermission`. **Mögliche Werte:** Zeichenkette mit dem Namen Objekt-Namen müssen den folgenden Namenskonventionen entsprechen: • Objekt-Namen können eine beliebige Folge von gültigen Unicode-Zeichen mit einer Länge zwischen 1-1024 Bytes enthalten, wenn sie UTF-8-kodiert sind. • Objekt-Namen dürfen keine Zeilenumbrüche oder Zeilenumbruch-Zeichen enthalten. • Objekt-Namen dürfen nicht mit `.well-known/acme-challenge` beginnen. • Objekt-Namen dürfen nicht `.` oder `..` benannt werden. *Beispiele*: • *`document`* • *`folder/subfolder/document`* • *`document.txt`*
`targetObjectName`	Für Operationen, die ein Objekt kopieren/verschieben/umbenennen. Der Parameter unterliegt denselben Beschränkungen wie der Parameter `objectName`.

`fileLocation`	Pfad der Datei, die in den Google-Cloud-Speicher hochgeladen werden soll; Dieser Parameter ist nur für die Operation `Upload` relevant **Mögliche Werte:** • System-Pfad, z. B. *`C:\X4\V_5.9\Server\X4DB\1\ExampleProcess\GoogleCloudStorageObject/Data/TextDocument.txt`* • X4db-Pfad, z. B. *`x4db://1/ExampleProcess/GoogleCloudStorageObject/Data/TextDocument.txt`* • Xstore-Pfad, z. B. *`xstore://ExampleProcess/GoogleCloudStorageObject/Data/TextDocument.txt`*
`contentType`	Inhalts-Typ des hochzuladenden Objekts **Mögliche Werte:** Zeichenkette mit einem gültigen Mime-Typ für das hochzuladende Objekt, z. B. *`text/plain`*, *`application/xml`* etc.
`overwrite`	Gibt an, ob während einer Operation die ein Objekt oder Bucket verschiebt/kopiert/umbenennt, Buckets oder Objekte mit demselben Namen überschrieben werden. Wenn diese Option nicht gewählt ist und am Zielort Objekte/Buckets mit demselben Namen existieren, enthält der Output eine Fehlermeldung. **Mögliche Werte:** • *`Checked`*: Vorhandene Objekte/Buckets werden überschrieben. • *`Unchecked`*: Vorhandene Objekte/Buckets werden nicht überschrieben, der Output enthält eine Fehlermeldung.

`permissionEntity`	Typ der ACL-Erlaubnis, die erstellt wird. **Mögliche Werte:** Auswahl mit folgenden Werten • `Project`: Spezifiziert eine Erlaubnis auf Basis einer Liste von Projektmitgliedern. Der Parameter `permissionProjectRole` muss ebenfalls angegeben werden, um festzulegen, aus welcher Unterliste die Mitglieder gewählt werden sollen. • `User`: Spezifiziert eine Erlaubnis auf Basis einer bestimmten E-Mail-Adresse, eine Erlaubnis für öffentlichen Zugriff oder eine Erlaubnis für alle Google-Konten. • `Domain`: Spezifiziert eine Erlaubnis auf Basis einer Internetdomäne, die mit G Suite oder Cloud Identity verbunden ist. • `Group`: Spezifiziert eine Erlaubnis auf Basis einer Gruppe von Google-Benutzern. Verwendet eine bestimmte E-Mail-Adresse, die mit einer Google-Gruppe verbunden ist.
`permissionRole`	Spezifiziert, ob die Erlaubnis einen Lese- oder einen Schreibzugriff gewährt. **Mögliche Werte:** Auswahl mit folgenden Werten • `READER`: Lesezugriff für Objekt/Bucket • `WRITER`: Lese- und Schreibzugriff für Objekt/Bucket • `OWNER`: Verwalten des Objekts/Buckets. Enthält die Rechte von `WRITER`.

`permissionName`	Angabe eines bestimmten Wertes, der in jeder Art von Berechtigung verwendet wird.
	
	Mögliche Werte: abhängig von der Angabe für `permissionEntity`
	
	• `permissionEntity="Project"`: Projektnummer. Erzeugt eine Zugriffserlaubnis für die diesem Projekt zugeordneten `OWNERS`, `EDITORS` und `VIEWERS`.
	• `permissionEntity="User"`:
	  ◦ `allUsers`: Erzeugt eine öffentliche Zugriffserlaubnis
	  ◦ `allAuthenticatedUsers`: Erzeugt eine Zugriffserlaubnis für alle Google-Konten
	  ◦ `<email address>`: Erzeugt eine Zugriffserlaubnis für den Benutzer, der mit dieser E-Mail-Adresse registriert ist.
	• `permissionEntity="Domain"`: `<USERNAME>@<YOUR_DOMAIN>.com`: Erzeugt eine Zugriffserlaubnis für Inhaber einer E-Mail-Adresse, die einer bestimmten Internetdomäne zugeordnet sind.
	• `permissionEntity="Group"`: E-Mail-Adresse der Google-Gruppe. Erzeugt eine Zugriffserlaubnis für alle Mitglieder dieser Google-Gruppe. Weitere Informationen zu Google-Gruppen.

`permissionProjectRole`	Parameter wirkt sich auf die Parameter `permissionName` und `permissionEntity`, wenn `permissionEntity="Project"`. **Mögliche Werte:** Auswahl mit folgenden Werten • `OWNERS` • `EDITORS` • `VIEWERS`

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters kam es zu einem Fehler.
`0`	• Operation `Upload`: Das hochzuladende Objekt existiert bereits • Operation `Download`: Das herunterzuladende Objekt existiert nicht • Operation `Delete`: Das zu löschende Objekt oder Bucket existiert nicht

Input

Der Adapter erwartet für manche Operationen ein beliebiges Input-Dokument (Binär- oder Base64-basiertes Dokument):

- Operation `Upload` : Ist der Parameter `fileLocation` angegeben, wird kein I nput-Dokument vorausgesetzt. Andernfalls wird ein Input vorausgesetzt.
- Alle anderen Operationen setzen kein Input-Dokument voraus.

Ausgabe

Je nach Operation gibt der Adapter verschiedene Ergebnisse aus:

- Operation `Upload` :
  - Wurde die Adapter-Operation erfolgreich ausgeführt (Status `1`), gibt der Adapter das Input-Dokument zurück.
  - Kam es bei der Ausführung des Adapters zu einem Fehler (Status `-1`), gibt der Adapter ein neues Dokument mit den Fehler-Details aus.
  - Existiert das hochzuladende Objekt bereist (Status `0`), entspricht die Ausgabe dem Input.
- Operation `Search`:
  - Wurde die Adapter-Operation erfolgreich ausgeführt (Status `1`), gibt der Adapter ein XML-Dokument mit allen Buckets und ihren Objekten aus:

    Beispiel-Ausgabe für die Operation Search

```
<?xml version="1.0" encoding="UTF-8"
<ListObjects>
    <Bucket name="first_bucket">
        <Object contentType="text/plain">example/first_blob</Object>
<Object contentType="text/plain">file.xml</Object>
        <Object contentType="application/x-www-form-
urlencoded;charset=UTF-8">newFolder</Object>
        <Object contentType="text/xml">newFolder/xml1 (1).xml</Object>
<Object contentType="text/plain">new_object</Object>
<Object contentType="application/x-www-form-
urlencoded;charset=UTF-8">otherFolder</Object
    </Bucket>
    <Bucket name="second_bucket">
        <Object contentType="text/plain">text/Hello</Object>
    </Bucket>
    <Bucket name="x4_upload_test_bucket">
        <Object contentType="image/png">Logo.png</Object>
        <Object contentType="application/xml">inputDocumentX4db.xml</
Object>
        <Object contentType="text/plain>testUploadProcess/Input/textFile</
Object>
<Object contentType="application/xml">testUploadProcess/Input/
xmlFile.xml</Object>
        <Object>wordDocument2</Object>
        <Object contentType="application/vnd.openxmlformats-
officedocument.wordprocessingml.document>wordDocumentFromLocationFile</
Object
    </Bucket>
</ListObjects>
```

  - Kam es bei der Ausführung des Adapters zu einem Fehler (Status `-1`), gibt der Adapter ein neues Dokument mit den Fehler-Details aus.
- Operation `Download`:

    - Wurde die Adapter-Operation erfolgreich ausgeführt (Status `1`), ist die Ausgabe das heruntergeladene Objekt (Text, XML, Binär-Dokument etc.).
    - Kam es bei der Ausführung des Adapters zu einem Fehler (Status `-1`), gibt der Adapter ein neues Dokument mit den Fehler-Details aus.
    - Existiert der Blob oder Container nicht (Status `0`), entspricht die Ausgabe dem Input.
- Operation `Delete`:
    - Wurde die Adapter-Operation erfolgreich ausgeführt (Status `1`), entspricht die Ausgabe dem Input.
    - Kam es bei der Ausführung des Adapters zu einem Fehler (Status `-1`), entspricht die Ausgabe dem Input.
    - Existieren der zu löschende Bucket oder das Objekt nicht (Status `0`), entspricht die Ausgabe dem Input.
- Operation `CopyObject`, `MoveOrRenameBucket`, `MoveOrRenameObject`, `UpdateObjectStorageClass`:
    - Bei regulärer Ausführung kein Output. Wenn die Operation nicht ausgeführt werden kann, wird eine Fehlermeldung ausgegeben.

10.3.5 Google Drive Connector

`Google Drive Connector`: Stellt eine Verbindung zum Google Drive Cloud-Service her und bietet Cloud-Storage-Datenverwaltungsfunktionen.

> ⚠ Google Drive erlaubt es, mehrere Objekte mit dem gleichen Namen zu haben. Beim Ausführen von Operationen, die sich auf mehrfach vorhandene Objekte beziehen, wird aktuell eine Fehlermeldung ausgegeben und die Operation wird nicht ausgeführt.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `CopyFile`: Kopiert eine bereits vorhandene Datei innerhalb des Google-Drive-Kontos • `CreateComment`: Legt einen Kommentar an • `CreateFolder`: Legt einen neuen Ordner im Google-Drive-Konto an • `DeleteFolder`: Löscht einen bereits vorhandenen Ordner • `DeleteFile`: Löscht eine bereits vorhandene Datei • `ExtractArchiveToFolder`: Extrahiert Dateien innerhalb eines ZIP-Ordners in einen Ordner • `GetFileContent`: Ruft die Inhalte eines bereits vorhandenen Dokuments ab • `GetFileMetadata`: Ruft die Metadaten einer Datei ab • `ListFiles`: Führt alle Dateien und Ordner des Wurzelverzeichnisses auf • `ListFilesInFolder`: Führt alle Dateien und Ordner innerhalb eines angegebenen Ordners auf • `ReadComments`: Liest bereits vorhandene Kommentare aus • `RevokePermission`: Widerruft Berechtigung gemäß der Parameterkonfiguration • `SetPermission`: Legt an oder aktualisiert Berechtigungen gemäß der Parameterkonfiguration • `UpdateFile`: Aktualisiert den Inhalt oder den Namen einer Datei • `UpdateFolderName`: Aktualisiert den Namen eines bereits vorhandenen Ordners • `UploadFile`: Lädt eine Datei in das Google-Drive-Konto hoch

Parameters

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.google.cloud.drive.GoogleCloudDriveAdapter*: Hauptklasse (Standard)
`serviceAccountFile`	Pfad der Servicekontodatei für die Authentifizierung **Mögliche Werte:** Zeichenkette mit dem Pfad der Servicekontodatei
`folderPath`	Ordnerpfad, in dem die extrahierte Datei abgelegt wird • Ist der Pfad nicht spezifiziert, wird der Pfad des Wurzelverzeichnisses verwendet. • Dieser Parameter wird für die Operationen `ListFilesInFolder`, `DeleteFile`, `GetFileContent` und `GetFileMetadata` verwendet. **Mögliche Werte:** Zeichenkette mit dem Ordnerpfad; Die Pfadangabe muss mit einem Schrägstrich `/` enden.

`sourcePath`	Pfad der Quelldatei Dieser Parameter wird für die Operationen `UploadFile`, `UpdateFile`, `CopyFile` und `ExtractArchiveToFolder` verwendet . **Mögliche Werte:** Zeichenkette mit dem Pfad zur Quelldatei
`destinationPath`	Pfad zum Zielordner oder zur Zieldatei Dieser Parameter wird für die Operationen `CopyFile`, `UploadFile`, `UpdateFile` und `ExtractArchiveToFolder` verwendet. **Mögliche Werte:** Zeichenkette mit dem Pfad zur Zieldatei oder zum Zielordner; Die Pfadangabe muss mit einem Schrägstrich `/` enden.
`fileName`	Dateiname – In einigen Fällen wird der Parameter zum Setzen oder Ändern des Dateinamen verwendet. Dieser Parameter wird für die Operationen `UploadFile`, `DeleteFile`, `GetFileContent`, `GetFileMetadata`, `UpdateFile` und `CopyFile` verwendet. **Mögliche Werte:** Zeichenkette mit dem Dateinamen
`folderName`	Ordnername • Für die Operation `ExtractArchiveToFolder` wird der Parameter verwendet, um einen neuen Ordner anzulegen, in dem die Dateien aus dem ZIP-Ordner entpackt werden. • Der Parameter wird zudem für die Operationen `CreateFolder`, `UpdateFolderName` und `DeleteFolder` verwendet. **Mögliche Werte:** Zeichenkette mit dem Ordnernamen
`comment`	Kommentar Dieser Parameter wird für *create*- und *read*-Operationen verwendet. **Mögliche Werte:** Zeichenkette mit dem Kommentar

`overwrite`	Angabe, ob bereits vorhandene Dateien überschrieben werden sollen Dieser Parameter wird außer für die Operation `CreteFolder` für alle *create*-, *copy*- oder *update*-Operationen verwendet. Für `false` wird eine a Meldung ausgegeben, wenn die Datei/der Ordner bereits vorhanden ist. **Mögliche Werte:** Boolescher Wert `true` oder `false`.
`scope`	Geltungsbereich der Berechtigung Dieser Parameter wird für die Operation `SetPermission` verwende t. **Mögliche Werte:** • `DOMAIN` • `GROUP` • `USER` • `ANYONE`
`role`	Rolle der Berechtigung Dieser Parameter wird für die Operation `SetPermission` verwende t. **Mögliche Werte:** • `FILE_ORGANIZER` • `READER` • `OWNER` • `WRITER` • `COMMENTER`
`account`	Domain oder E-Mail-Adresse des Benutzers oder der Gruppe Dieser Parameter wird für die Operationen `SetPermission` und `RevokePermission` verwendet. **Mögliche Werte:** Zeichenkette mit der Domain oder E-Mail-Adresse

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt
`-1`	Bei der Ausführung des Adapters kam es zu einem Fehler

Input

Der Adapter wird über Parameter verwaltet. Für einige Operationen wie `ExtractArchiveToFolder`, `UploadFile` oder `UpdateFile` kann jedoch ein Input übergeben werden, der den hochzuladenden oder zu aktualisierenden Inhalt darstellt. Wenn bei diesen Operationen kein Input übergeben wird, wird der Parameter `sourcePath` verwendet.

Ausgabe

Der Adapter gibt folgende Struktur aus, wenn eine *list*-Operation verwendet wird:

Ausgabe für list-Operationen

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Files>
	<File FullName="X4-ADK.pdf" id="1_nNnJVwCyNXrFWiGDok1GNKmdZeAcXK7"/>
	<File FullName="X4-Activities.pdf" id="1soJS2I4JgG4pNxypbI9MnTKSON1iLWcI"/>
	<File FullName="X4-Release-Notes.pdf" id="1C8dh8I6r2LDsPP3avdkYNIVLCGmvf54T"/>
	<File FullName="X4-ESB.pdf" id="1lPatZJT2yxI1IdfzlYy8hINO56TOhCdC"/>
	<File FullName="X4-BPM.pdf" id="1R5_mJ4x6_qcPZhrnaP-z_zoFd6kJJmAn"/>
	<File FullName="X4-Administration-Guide.pdf" id="1di3dhqUe3bLXx8OWQqtqcBLcIeXPS4N-"/>
</Files>
```

Für die Operation `GetFileMetadata` werden folgende Informationen ausgegeben:

Ausgabe für die Operation GetFileMetadata

```
<?xml version="1.0" encoding="UTF-8"?>
<File FullName="X4-Release-Notes-Comment.pdf" Id="18upYSTk8pkGMm8YVT46LHa9R9myYzf7I">
    <Kind>drive#file</Kind>
    <Parents>
        <Id>0AOvK1n-M5IHmUk9PVA</Id>
    </Parents>
    <Permissions>
        <Permission>
            <Id>03478937985708787222</Id>
            <Account>service-account@shaped-fx-259808.iam.gserviceaccount.com</Account>
            <Scope>user</Scope>
            <Role>owner</Role>
        </Permission>
    </Permissions>
    <Created>2020-01-07T10:45:33.338</Created>
    <Size>4376271</Size>
</File>
```

10.3.6 Microsoft Azure Blob Storage Connector

`Microsoft Azure Blob Storage Connector`: Ermöglicht die Anbindung an einen Microsoft Azure Blob Storage.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** ⓘ Die Parameter `containerName` und `blobName` sind für alle Operationen erforderlich. • `Search`: Sucht nach Informationen (Container oder Blobs) im Microsoft-Azure-Konto • `Append`: Fügt Informationen zum ausgewählten Blob im Microsoft-Azure-Konto hinzu **Erforderliche Parameter:** `fileLocation` • `Upload`: Lädt Container oder Blobs hoch oder legt Container oder Blobs im Microsoft-Azure-Konto an **Erforderliche Parameter:** `fileLocation` • `Download`: Lädt ein Blob vom Microsoft-Azure-Konto herunter • `Delete`: Löscht ein Container oder Blob vom Microsoft-Azure-Konto • `UpdateDirectory`: Aktualisiert den Namen des Quellverzeichnisses mit dem Namen des Zielverzeichnisses **Erforderliche Parameter:** `sourceDirectoryName`, `targetDirectoryName` • `MoveDirectory`: Verschiebt den Inhalt vom Quellverzeichnis in das Zielverzeichnis **Erforderliche Parameter:** `sourceDirectoryName`, `targetDirectoryName` • `DeleteDirectory`: Löscht den Inhalt des Quellverzeichnisses **Erforderliche Parameter:** `sourceDirectoryName` , `targetDirectoryName` • `MoveBlobDirectory`: Verschiebt einen bestimmten Blob vom Quellverzeichnis in das Zielverzeichnis **Erforderliche Parameter:** `sourceDirectoryName` , `targetDirectoryName`

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.microsoft.azure.blob.MicrosoftAzureBlobStorageAdapter*: Hauptklasse (Standard)
`accountName`	Name des Microsoft-Azure-Kontos **Mögliche Werte:** Beliebige Zeichenkette, z. B. *`softproject`*
`accountKey`	Passwort für das Microsoft-Azure-Konto **Mögliche Werte:** Beliebige Zeichenkette mit dem Schlüssel z. B. *`AD9ohFhJ6Hve9ODJEXTw53MjR1tjoUZ8lWfDlw5UFqGMih3VtUCapIwIfEMBmITeShQN9MbCITE4SzxTcGIybw==`*

`containerName`	Name des Containers in dem die Blobs organisiert werden **Mögliche Werte:** Beliebige Zeichenkette mit dem Namen Container-Namen müssen folgende Namenskonventionen erfüllen: • Container-Namen müssen mit einem Buchstaben oder einer Zahl beginnen und können nur Buchstaben, Zahlen oder Bindestriche (-) enthalten • Jeder Bindestrich (-) muss unmittelbar vor und nach einem Buchstaben oder einer Zahl stehen; aufeinanderfolgende Bindestriche sind in Container-Namen nicht zulässig • Container-Namen dürfen nur Kleinbuchstaben enthalten • Container-Namen müssen zwischen 3 und 63 Zeichen enthalten
`blobName`	Name der gespeicherten Blobs **Mögliche Werte:** Beliebige Zeichenkette mit dem Blob-Namen Blob-Namen müssen folgende Namenskonventionen erfüllen: • Blob-Namen können eine beliebige Kombination aus Zeichen enthalten • Blob-Namen müssen mindestens ein Zeichen und dürfen nicht mehr als 1.024 Zeichen lang sein • Bei Blob-Namen wird zwischen Groß- und Kleinbuchstaben unterschieden • Reservierte URL-Zeichen müssen ordnungsgemäß maskiert sein • Die Anzahl der Pfad-Segmente, die den Blob-Namen umfassen, darf 254 nicht überschreiten. Ein Pfad-Segment ist die Zeichenkette zwischen aufeinanderfolgenden Begrenzungszeichen (z. B. der Schrägstrich '/'), die dem Namen eines virtuellen Verzeichnisses entsprechen.
`fileLocation`	Pfad der hochzuladenden Datei; Dieser Parameter ist nur für die Operation `Upload` gültig. **Mögliche Werte:** • System-Pfad, z. B. *`C:\X4\V_5.9\Server\X4DB\1\ExampleProcess/MAzureBlobStore/Data/TextDocument.txt`* • X4db-Pfad, z. B. *`x4db://1/ExampleProcess/MAzureBlobStore/Data/TextDocument.txt`* • Xstore-Pfad, z. B. *`xstore://ExampleProcess/MAzureBlobStore/Data/TextDocument.txt`*
`overwrite`	Information, ob Blobs überschrieben werden oder nicht
`sourceDirectoryName`	Name des Quellverzeichnisses für alle `Directory`-Operationen **Mögliche Werte:** Beliebige gültige Zeichenkette mit einem Abschließenden `/`, z. B. *`sourceFolder/`*
`targetDirectoryName`	Name des Zielverzeichnisses für alle `Directory` -Operationen **Mögliche Werte:** Beliebige gültige Zeichenkette mit einem Abschließenden `/` , z. B. *`targetFolder/`*

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt
`-1`	Es kam zu einem Fehler während der Ausführung

`0`	• Operation `Download`: Der Blob, der heruntergeladen werden soll, existiert nicht
	• Operation `Delete`: Der Blob, der gelöscht werden soll, existiert nicht
	• Operationen `UpdateDirectory`, `MoveDirectory`, `DeleteDirectory`, `MoveBlobDirectory`:
	  • das Quellverzeichnis existiert nicht
	  • das Zielverzeichnis konnte nicht gefunden werden (Operationen `MoveDirectory` , `MoveBlobDirectory`)
	  • der Blob existiert nicht im Quellverzeichnis (Operation `MoveBlobDirectory`)

Input

Der Adapter erwartet ein beliebiges Input-Dokument, z. B. ein Binär- oder ein Base64-Dokument (XML oder Text).

- Operation `Upload`: Ist der Parameter `fileLocation` angegeben, so kann auf das Input-Dokument verzichtet werden. In allen anderen Fällen wird ein Input-Dokument vorausgesetzt.
- Für alle anderen Operationen wird kein Input-Dokument vorausgesetzt.

Output

Abhängig von der gewählten Operation, gibt der Adapter verschiedene Dokumente aus:

Operation `Upload`:

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt der Adapter ein XML-Dokument mit der URL der hochgeladenen Datei oder dem Dokument aus.

  Operation Upload: Status 1

  ```
  <?xml version="1.0" encoding="UTF-8"
  <BLOB>
      <Upload>https://storage_account_name.blob.core.windows.net/container_name/
  blob_name</Upload>
  </BLOB>
  ```

- Bei einem Fehler (Status `-1`), entspricht die Ausgabe dem Input-Dokument.

Operation `Search`:

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt der Adapter ein XML-Dokument mit allen Containern und ihren Blob-URI-Objekten aus.

Operation Search: Status 1

```
<?xml version="1.0" encoding="UTF-8"
<ListBlobs>
    <Container name="pc-files">
        <Blob name="C-structure.xml">https://softproject.blob.core.windows.net/
pc-files/C-structure.xml</Blob>
        <Blob name="XStore_WordDocument.doc">https://
softproject.blob.core.windows.net/pc-files/XStore_WordDocument.doc</Blob>
        <Blob name="jar_files(1).zip">https://
softproject.blob.core.windows.net/pc-files/jar_files(1).zip</Blob>
        <Blob name="x4db/HipChat-4.30.6.1676.exe">https://
softproject.blob.core.windows.net/pc-files/x4db/HipChat-4.30.6.1676.exe</Blob>
        <Blob name="xstore/MWDocument.doc">https://
softproject.blob.core.windows.net/pc-files/xstore/MWDocument.doc</Blob>
    </Container>
    <Container name="testcontainer-prototype">
        <Blob name="base64Files/file1">https://
softproject.blob.core.windows.net/testcontainer-prototype/base64Files/file1</
Blob>
        <Blob name="documents/doc1">https://softproject.blob.core.windows.net/
testcontainer-prototype/documents/doc1</Blob>
        <Blob name="documents/doc1.docx">https://
softproject.blob.core.windows.net/testcontainer-prototype/documents/doc1.docx</
Blob>
        <Blob name="imagesFiles/image1">https://
softproject.blob.core.windows.net/testcontainer-prototype/imagesFiles/image1</
Blob>
        <Blob name="textFiles/subTextFiles/text5.txt">https://
softproject.blob.core.windows.net/testcontainer-prototype/textFiles/
subTextFiles/text5.txt</Blob>
        <Blob name="xml1.xml">https://softproject.blob.core.windows.net/
testcontainer-prototype/xml1.xml</Blob>
        <Blob name="xmlFiles/xml1">https://softproject.blob.core.windows.net/
testcontainer-prototype/xmlFiles/xml1</Blob>
        <Blob name="xmlFiles/xml1.xml">https://
softproject.blob.core.windows.net/testcontainer-prototype/xmlFiles/xml1.xml</
Blob>
    </Container>
    <Container name="test">... </Container>
</ListBlobs>
```

- Bei einem Fehler (Status `-1`), entspricht die Ausgabe dem Input-Dokument.

Operation `Download`:

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt der Adapter den heruntergeladenen Blob (Text, XML, Binärdokument etc.) aus.
- Bei einem Fehler (Status `-1`), entspricht die Ausgabe dem Input-Dokument.
- Existieren Blob oder Container nicht (Status 0), entspricht die Ausgabe dem Input-Dokument.

Operation
UpdateDirectory
:

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt das Output-Dokument nach dem Aktualisieren des Quellordners mit dem Namen des Zielordners alle betroffenen Blobs aus.

Operation UpdateDirectory: Status 1

```
<?xml version="1.0" encoding="UTF-8"
<UpdateDirectory targetDirectory="testFolderR/" sourceDirectory="testFolder/">
    <Blob newName="testFolderR/C-structure.xml" name="testFolder/C-
structure.xml"/>
    <Blob newName="testFolderR/X4_1_Configuration.txt" name="testFolder/
X4_1_Configuration.txt"/>
</UpdateDirectory>
```

- Bei einem Fehler (Status `-1`), entspricht die Ausgabe dem Input-Dokument.
- Existiert das Quellverzeichnis nicht (Status `0`), ist das Output ein leeres XML-Dokument.

Operation UpdateDirectory: Status 0

```
<?xml version="1.0" encoding="UTF-8"
<UpdateDirectory targetDirectory="otherFiles/testSubFolderR/" sourceDirectory="
textFiles/subTextFiles"/>
```

Operation `MoveDirectory`:

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt das Output-Dokument nach dem Verschieben des Quellordners in den Zielordner alle betroffenen Blobs aus.

Operation MoveDirectory: Status 1

```
<?xml version="1.0" encoding="UTF-8"
<MoveDirectory targetDirectory="textFiles/" sourceDirectory="testFolderR/">
    <Blob newName="textFiles/C-structure.xml" name="testFolderR/C-
structure.xml"/>
    <Blob newName="textFiles/X4_1_Configuration.txt" name="testFolderR/
X4_1_Configuration.txt"/>
</MoveDirectory>
```

- Bei einem Fehler (Status `-1`), entspricht die Ausgabe dem Input-Dokument.
- Existiert das Quell- oder Zielverzeichnis nicht (Status `0`), ist das Output ein leeres XML-Dokument.

Operation MoveDirectory: Status 0 - Zielverzeichnis existiert nicht

```
<?xml version="1.0" encoding="UTF-8"
<MoveDirectory targetDirectory="otherFiles/testSubFolderR/" sourceDirectory="te
xtFiles/subFolderR"> Target Directory not found for Move operation </
MoveDirectory>
```

Operation MoveDirectory: Status 0 - Quellverzeichnis existiert nicht

```
<?xml version="1.0" encoding="UTF-8"
<MoveBlobDirectory targetDirectory="textFiles/" sourceDirectory="otherFiles/
testSubFolderR"/>
```

Operation `DeleteDirectory`:

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt das Output-Dokument nach dem Löschen des Quellordners und seines Inhalts alle betroffenen Blobs aus.

Operation DeleteDirectory: Status 1

```
<?xml version="1.0" encoding="UTF-8"
<DeleteDirectory targetDirectory="" sourceDirectory="imagesFiles2/">
    <Blob newName="" name="imagesFiles2/image1.png"/>
</DeleteDirectory>
```

- Bei einem Fehler (Status `-1`), entspricht die Ausgabe dem Input-Dokument.
- Existiert das Quellverzeichnis nicht (Status `0`), ist das Output ein leeres XML-Dokument.

Operation DeleteDirectory: Status 0

```
<?xml version="1.0" encoding="UTF-8"
<DeleteDirectory targetDirectory="" sourceDirectory="imagesFiles2/"/>
```

Operation `MoveBlobDirectory`:

- Wurde die Operation erfolgreich ausgeführt (Status `1`), gibt das Output-Dokument alle Blobs aus, die vom Quell- in den Zielordner verschoben wurde.

Operation MoveBlobDirectory: Status 1

```
<?xml version="1.0" encoding="UTF-8"
<MoveBlobDirectory targetDirectory="/" sourceDirectory="textFiles/">
    <Blob newName="X4_1_Configuration.txt" name="textFiles/
X4_1_Configuration.txt"/>
</MoveBlobDirectory>
```

- Bei einem Fehler (Status `-1`), entspricht die Ausgabe dem Input-Dokument.
- Existieren Quell- oder Zielverzeichnis nicht (Status `0`), ist das Output ein leeres XML-Dokument.

Operation MoveBlobDirectory: Status 0 - Zielverzeichnis existiert nicht

```
<?xml version="1.0" encoding="UTF-8"
<MoveBlobDirectory targetDirectory="otherFiles/testSubFolderR/" sourceDirectory="
textFiles/subFolderR"> Target Directory not found for Move operation </
MoveDirectory>
```

Operation MoveBlobDirectory: Status 0 - Quellverzeichnis existiert nicht

```
<?xml version="1.0" encoding="UTF-8"
<MoveBlobDirectory targetDirectory="textFiles/" sourceDirectory="otherFiles/
testSubFolderR"/>
```

10.3.7 Microsoft Azure Table Storage Connector

`Microsoft Azure Table Storage Connector`: Ermöglicht die Verbindung zu Microsoft Azure, einer Cloud-Computing-Plattform von Microsoft, die Cloud Cosmos DB Table Storage zur Verfügung stellt.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `CreateTable`: Legt eine neue Tabelle an **Erforderliche Parameter:** `tableName` • `ListTable`: Listet alle Tabellen auf • `DeleteTable`: Löscht eine bereits vorhandene Tabelle **Erforderliche Parameter:** `tableName` • `InsertEntity`: Fügt die Entität ein, wenn die Entität noch nicht vorhanden ist, oder ersetzt eine bestehende Entität. Nachdem die Operation erfolgreich ausgeführt wurde, enthält die Tabelle die neue Entität mit den definierten Eigenschaften und die vorherige Entität und ihre Eigenschaften werden ersetzt. **Erforderliche Parameter:** `tableName`, `partitionKey`, `rowKey` • `InsertBatchEntities`: Fügt mehrere Entitäten mit einem Schreibvorgang ein, oder ersetzt bestehende Entitäten. Nutzt die `InsertOrReplace`-Entität. **Erforderliche Parameter:** `tableName` • `RetrieveEntity`: Ruft eine einzelne, spezifizierte Entität ab **Erforderliche Parameter:** `tableName`, `partitionKey`, `rowKey` • `RetrieveAllEntities`: Ruft alle Entitäten von einem bestimmten `partitionKey` ab. **Erforderliche Parameter:** `tableName`, `partitionKey` • `RetrieveRangeEntities`: Ruft eine Reihe von Entitäten mit dem `partitionKey` ab und filtert nach einer Reihe von `rowKey`-Werten. Der `partitionKey`-Wert identifiziert eine bestimmte Partition und die `rowKey`-Werte identifizieren eine Teilmenge der Entitäten dieser Partition. **Erforderliche Parameter:** `tableName`, `partitionKey`, `rowKey`, `filterCondition` • `DeleteEntity`: Löscht eine einzelne Entität. **Erforderliche Parameter:** `tableName`, `partitionKey`, `rowKey`

	• `ModifyEntitiy`: Ändert eine Entität und wählt zwischen einer Zusammenführen- und einer Ersetzen-Operation mit der Parameterprüfung `replaceExistingItems`. Wenn die Ersetzen-Operation ausgewählt ist (`replaceExistingItems="true"`), dann bleiben nach der Operation nur die angegebenen Eigenschaften erhalten, alle anderen Eigenschaften werden gelöscht. Wenn die Zusammenführen-Operation gewählt ist (`replaceExistingItems="false"`), dann bleiben alle Eigenschaften erhalten und die bereits vorhandenen werden aktualisiert. **Erforderliche Parameter:** `tableName`, `partitionKey`, `rowKey`, `replaceExistingItems` • `QuerySubsetOfEntityProperties`: Ruft eine Teilmenge von Eigenschaften der Entitäten gemäß eines XML-Inputs ab. Wenn `rowKey` und `partitionKey` angegeben sind, dann werden nur die passenden Eigenschaften der Entitäten abgerufen. Wenn weder `rowKey` noch `partitionKey` angegeben sind, dann werden alle Eigenschaften der Tabellenentitäten abgerufen. Die Ausgabe dieser Operation ist eine XML-Datei, die die Teilmenge der Eigenschaften innerhalb eines übergeordneten Elements für jede zugehörige Entität enthält. **Erforderliche Parameter:** `tableName`, `partitionKey`, `rowKey`

Parameter

`Adapter`	Adapter Hauptklasse (nicht ändern!) **Mögliche Werte:** *de.softproject.integration.adapter.microsoft.azure.table.MicrosoftAzureTableStorageAdapter*: Hauptklasse (Standard)
`accountName`	Name des Microsoft Azure Kontos **Mögliche Werte:** Beliebige gültige Zeichenkette, z. B. `softproject`
`accountKey`	Passwort zum Microsoft Azure Konto **Mögliche Werte:** Beliebige Zeichenkette mit dem von Microsoft Azure bereitgestellten Schlüssel, z. B. *AD9ohFhJ6Hve9ODJEXTw53MjR1tjoUZ8lWfDlw5UFqGMih3VtUCapIwIfEMBmITeShQN9MbCITE4SzxTcGIybw==*
`endPoint`	Azur Cosmos DB Endpunkt

`tableName`	Name der Tabelle, in der die Entitäten gespeichert sind **Mögliche Werte:** Zeichenkette mit der ID für die neue Tabelle Tabellen-Namen müssen die gleichen Namenskonventionen erfüllen wie Datenbank-IDs. Datenbank-Namen müssen zwischen 1 und 255 Zeichen lang sein und dürfen weder `/ \ # ?` noch ein Leerzeichen enthalten.
`partitionKey`	Element des Primärschlüssels Entitäten mit dem gleichen Partitionsschlüssel können schneller abgefragt und in atomaren Operationen eingefügt/aktualisiert werden. **Mögliche Werte:** Beliebige Zeichenkette, die den Partitionsschlüssel enthält
`rowKey`	Element des Primärschlüssels Der Zeilenschlüssel einer Entität ist ihre eindeutige Kennung innerhalb einer Partition. **Mögliche Werte:** Beliebige Zeichenkette, die den Zeilenschlüssel enthält
`filterCondition`	Vergleicht `partitionKey` und `rowKey` **Mögliche Werte:** • EQUAL • GREATER_THAN • GREATER_THAN_OR_EQUAL • LESS_THAN • LESS_THAN_OR_EQUAL • NOT_EQUAL
`replaceExistingItems`	Gibt an, ob `ModifyEntity` eine Zusammenführen- oder Ersetzen-Operation ausführt. Weitere Informationen auf http://azure.github.io/azure-storage-java/. **Mögliche Werte:** • `true`: Ersetzen-Operation ausführen • `false`: Zusammenführen-Operation ausführen

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt
`-1`	Bei der Ausführung des Adapters kam es zu einem Fehler
`0`	Operation `CreateTable`: Die Tabelle konnte nicht angelegt werden, aber es wurde weder ein Fehler noch eine Exception zurückgegeben. Operation `DeleteTable`: Er angegebene `tableName` existiert nicht.

Input

Der Adapter erwartet ein XML-Dokument als Input.

- Für die `InsertEntity`-Operation wird eine XML-Datei mit folgender Struktur erwartet:

  Erwartete Struktur für die Operation InsertEntity

  ```
  <?xml version="1.0" encoding="UTF-8"
  <Entity>
      <Properties1 name="Department" value="ITAdministration"/>
      <Properties2 name="Address" value="C\Pirandello"/>
      <Properties3 name="Name" value="Paolo"/>
      <Properties4 name="Surname" value="Green"/>
  </Entity>
  ```

  ⓘ Die Eigenschaft `name` darf kein geschütztes Wort wie beispielsweise *Id* sein.

- Für die `ModifyEntity`-Operation wird eine XML-Datei mit folgender Struktur erwartet, Properties mit gleichem Namen und Wert werden aktualisiert:

  Erwartete Struktur für die Operation ModifyEntity

  ```
  <?xml version="1.0" encoding="UTF-8" ?>
  <Entity>
      <Properties name="Name" value="Carlos" />
  </Entity>
  ```

- Für die `QueryOfSubsetEntityProperties`-Operation wird eine XML-Datei erwartet, in der die erforderlichen Properties spezifiziert sind:

  Erwartete Struktur für die Operation QueryOfSubsetEntityProperties

  ```
  <?xml version="1.0" encoding="UTF-8" ?>
  <Entity>
      <Property name="Department" />
      <Property name="Surname" />
      <Property name="nonExistingProperty" />
  </Entity>
  ```

Ausgabe

Der Adapter gibt je nach ausgeführter Operation unterschiedliche Arten von Dokumenten zurück:

- Ausgabe der `RetrieveEntity`-Operation:

Ausgabe für die Operation RetrieveEntity

```
<?xml version="1.0" encoding="UTF-8"
<Entity partitionKey="ITEmployee" rowKey="ref-0001">
    <Properties name="Department" value="ITAdministration"/>
    <Properties name="Address" value="C\Pirandello"/>
    <Properties name="Surname" value="Green"/>
    <Properties name="Name" value="Paolo"/>
</Entity>
```

- Ausgabe der `RetrieveAllEntities`- und `RetrieveRangeEntities`-Operation:

Ausgabe für die Operation RetrieveAllEntities und RetrieveRangeEntities

```
<?xml version="1.0" encoding="UTF-8" ?>
<ListEntities>
    <Entity partitionKey="ITEmployee" rowKey="ref-0001">
        <Properties name="Department" value="ITAdministration" />
        <Properties name="Address" value="C\Pirandello" />
        <Properties name="Surname" value="Green" />
        <Properties name="Name" value="Paolo" />
    </Entity>
    <Entity partitionKey="ITEmployee" rowKey="ref-0002">
        <Properties name="Department" value="Developments" />
        <Properties name="Address" value="C\Francisco" />
        <Properties name="Surname" value="Lopez" />
        <Properties name="Name" value="Luis" />
    </Entity>
    <Entity partitionKey="ITEmployee" rowKey="ref-0003">
        <Properties name="Department" value="Sales" />
        <Properties name="Address" value="C\Espada" />
        <Properties name="Surname" value="Red" />
        <Properties name="Name" value="Paula" />
    </Entity>
</ListEntities>
```

- Ausgabe der `QueryOfSubsetEntityProperties`-Operation:

Ausgabe für die Operation QueryOfSubsetEntityProperties

```
<?xml version="1.0" encoding="UTF-8" ?>
<ListEntities>
    <Entity>
        <Property partitionKey="ITEmployee" />
        <Property rowKey="ref-0001" />
        <Property name="Department" value="ITAdministration" />
        <Property name="nonExistingProperty" />
        <Property name="Surname" value="Green" />
    </Entity>
</ListEntities>
```

Die beiden Properties `partitionKey` und `rowKey` werden immer zurückgegeben, da mit ihnen die Entität eindeutig identifiziert werden kann. Da die Operation in der Lage ist, mehrere Properties von mehr als einem Objekt zurückzugeben, ist es wichtig, die Eigenschaften der dazugehörigen Entität zuzuordnen.

10.4 Communication

- AS/400 PCML
- File System
- File System Bulk
- FTP
- iCalendar
- Ping
- SCP
- TCP/IP Connector/Listener
- Telnet Connector

10.4.1 AS/400 PCML

`AS/400 PCML`: Ruft Programme auf *AS/400*-Systemen (*IBM eServer iSeries* bzw. *System i*) über die *Program Call Markup Language* (PCML) auf und gibt die Ergebnisse des ausgeführten Programms als XML-Dokument zurück.

Ehemalige Bezeichnung bis Version 4.5.1: *AS/400 PCML Connection Adapter*

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *Execute*: Funktions-Adapter ausführen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.PCMLAdapter*: Hauptklasse (Standard)
systemName	Host-Name des AS/400-Systems **Mögliche Werte:** Beliebiger Hostname (z. B. *localhost*)
userId	Benutzername für die Server-Authentifizierung **Mögliche Werte:** • Beliebige Zeichenkette • **CURRENT*: aktuelle Benutzer-ID aufrufen
password	Passwort für die Server-Authentifizierung **Mögliche Werte:** Beliebige Zeichenkette
encoding	Zeichenkodierung des Input- und Output-XML-Dokuments **Mögliche Werte:** Beliebige gültige Zeichenkodierung (z. B. *UTF-8*)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Adapter erwartet eine adapterspezifische Input-XML-Struktur, wobei jedes Element `<program>` einem Programm entspricht, jedes Element `<data>` darin ein Feld definiert und jedes Element `<structure>` als Datenstruktur-Element oder als ein Argument für ein Programm dienen kann:

```
<pcml version="1.0">
   <program>
      <data></data>
      <struct>
          <data></data>
      </struct>
   </program>
</pcml>
```

Ausgabe

Der erfolgreich ausgeführte PCML-Adapter gibt ein XML-Dokument mit folgendem Muster zurück:

```
<pcml>
   <program_name>
      <data_values></data_values>
      <struct_name1></struct_name1>
   </program_name>
</pcml>
```

10.4.2 File System

`File System`: Speichert, liest, löscht, kopiert, verschiebt, listet Dateien und Ordner oder fügt Dateien einer Zieldatei an, die sich im Dateisystem des X4 Servers oder auf verbundenen Windows- oder Samba-Netzlaufwerken befinden.

> ⚠ Der Adapter *File System* darf nicht verwendet werden, um Dateien im X4 Repository zu verändern, da er nicht auf das X4 Repository, sondern direkt auf das Dateisystem zugreift. Wird der Adapter hierfür verwendet, kann es zu Fehlermeldungen und Störungen kommen. Um Dateien im X4 Repository zu verändern, verwenden Sie den Adapter Projects bzw. Project Access.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Write File`: Daten aus dem vorangegangenen Prozess-Schritt in die im Parameter `path` angegebene Zieldatei schreiben. • `Append File`: Datei aus dem vorangegangenen Prozess-Schritt in die im Parameter `path` angegebene Zieldatei anhängen. ⓘ Wenn die im Parameter `path` festgelegte Zieldatei nicht existiert, wird eine Datei erstellt. • `Read File`: Im Parameter `path` festgelegte Datei lesen. • `Delete`: Im Parameter `path` festgelegte Dateien oder Ordner löschen. • `Copy`: Im Parameter `path` festgelegte Dateien oder Ordner in den im Parameter `targetPath` festgelegten Ordner kopieren. • `Move`: Im Parameter `path` festgelegte Dateien oder Ordner in den im Parameter `targetPath` festgelegten Ordner verschieben. • `Exists`: Prüfen, ob die im Parameter `path` festgelegten Dateien oder Ordner existieren. Gibt als Ergebnis eine XML mit Metadaten aus. • `Create Folder`: Im Parameter `path` festgelegten Ordner erstellen. • `List Folder`: Namen und Metadaten von Dateien und Ordnern aus dem im Parameter `path` festgelegten Ordner auflisten. Gibt als Ergebnis eine XML mit Metadaten aus. ⓘ Der Inhalt der Dateien wird nicht gelistet.

Parameter

`Adapter`	Adapter Hauptklasse (nicht ändern!) **Mögliche Werte:** • `de.softproject.integration.adapter.filesystem.FileSystemAdapter`: Hauptklasse (Standard)
`path`	Pfad zu einer Datei oder einem Ordner im Dateisystem des X4 Servers oder zu einem verbundenen Windows- oder Samba-Netzlaufwerk. Hierzu benötigt der X4 Server entsprechende Berechtigungen auf den Ordner. **Mögliche Werte:** • Beliebige Zeichenkette, ggf. inklusive Platzhaltern und Dateiendung z.B. *`%USER_COLLECTION%/Data/Input.xml`* • *`c:/`*: Auf die Windows-Systempartition C:\ zugreifen • Beliebige UNC-Netzlaufwerksangabe nach dem Schema `\\\\Server\\Freigabe\\Ordner\\` mit beliebigen Leer- und Sonderzeichen z.B. `\\\\SP-PM-01\\share\\Öffentlicher Ordner\\`

`targetPath`	Zusätzlicher Pfad zu einer Datei oder einem Ordner im Dateisystem des X4 Servers oder zu einem verbundenen Windows- oder Samba-Netzlaufwerk. Hierzu benötigt der X4 Server entsprechende Berechtigungen auf den Ordner. ⚠ Dieser Parameter wird nur bei der Operation `Move` und `Copy` verwendet. **Mögliche Werte:** • Beliebige Zeichenkette, ggf. inklusive Platzhaltern und Dateiendung z.B. *`%USER_COLLECTION%\Data\Input.xml`* • *`c:\`* Auf die Windows-Systempartition C:\ zugreifen • Beliebige UNC-Netzlaufwerksangabe nach dem Schema *`\\\\Server\\Freigabe\\Ordner\\`* mit beliebigen Leer- und Sonderzeichen z.B. *`\\\\SP-PM-01\\share\\Öffentlicher Ordner\\`*
`nameFilter`	Filtert den Output der Operation `ListFolder` nach Name **Mögliche Werte:** • Beliebige Zeichenkette (z.B. *`myFolder`*) • Platzhalter (Wildcard) `*`
`addNewline`	Fügt in der Zieldatei eine neue Zeile vor der anzuhängenden Datei ein. Kann nur mit der Operation `Append File` verwendet werden. **Mögliche Werte:** • `None`: (Standard) Fügt keine neue Zeile ein. • `Auto`: Fügt eine neue Zeile ein, wenn zuvor keine neue Zeile eingefügt wurde. Erkennt das Format einer bereits eingefügten neuen Zeile. • `Windows`: Fügt eine neue Zeile im Windows-Format ein. • `Unix`: Fügt eine neue Zeile im Unix-Format ein.

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

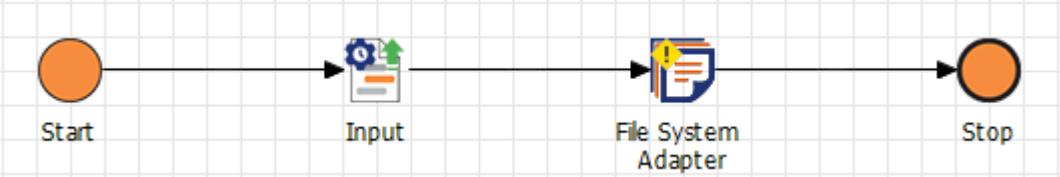
Weitere Informationen finden Sie im Abschnitt

- Operation: Write File
- Operation: Append File
- Operation: Read File
- Operation: Delete
- Operation: Copy
- Operation: Move
- Operation: Exists
- Operation: Create Folder
- Operation: List Folder

10.4.2.1 Operation: Write File

Die Operation `Write File` schreibt Daten aus dem vorangegangenen Prozess-Schritt in die in Parameter `path` angegebene Zieldatei.

10.4.2.1.1 Beispiele



Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Write File`, um die Daten aus dem vorangegangenen Prozess-Schritt in die in Parameter `path` angegebene Zieldatei zu schreiben.

Parameter	Wert
`path`	C:\MyFolder\File.xml

Weitere Informationen finden Sie im Abschnitt

- Operation: Write File
- Operation: Append File
- Operation: Read File
- Operation: Delete
- Operation: Copy
- Operation: Move
- Operation: Exists
- Operation: Create Folder
- Operation: List Folder

10.4.2.2 Operation: Append File

Die Operation `Append File` hängt die Daten aus dem vorangegangenen Prozess-Schritt in die in Parameter `path` angegebene Zieldatei.

> ⓘ Wenn die im Parameter `path` festgelegte Zieldatei nicht existiert, wird eine Datei erstellt.

10.4.2.2.1 Beispiele

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Append File`, um die Daten aus dem vorangegangenen Prozess-Schritt in die in Parameter `path` angegebene Zieldatei anzuhängen.

Parameter	Wert
`path`	C:\MyFolder\File.xml
`addNewline`	`Auto`

10.4.2.2.1.1 Zieldatei

```
<?xml version="1.0" encoding="UTF-8"?>
<Cars>
    <Car>
        <Brand>Audi</Brand>
        <Country>Germany</Country>
    </Car>
</Cars>
```

10.4.2.2.1.2 Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Cars>
    <Car>
        <Brand>BMW</Brand>
        <Country>Germany</Country>
    </Car>
    <Car>
        <Brand>Hyundai</Brand>
        <Country>South Korea</Country>
    </Car>
</Cars>
```

10.4.2.2.1.3 Output

```
<?xml version="1.0" encoding="UTF-8"?>
<Cars>
    <Car>
        <Brand>Audi</Brand>
        <Country>Germany</Country>
    </Car>
</Cars>
<?xml version="1.0" encoding="UTF-8"?>
<Cars>
    <Car>
        <Brand>BMW</Brand>
        <Country>Germany</Country>
    </Car>
    <Car>
        <Brand>Hyundai</Brand>
        <Country>South Korea</Country>
    </Car>
</Cars>
```

⚠ Bei dem Output handelt es sich um eine nicht valide XML-Datei. Der Output muss im Anschluss mithilfe eines Mappings in eine valide XML-Datei transformiert werden.

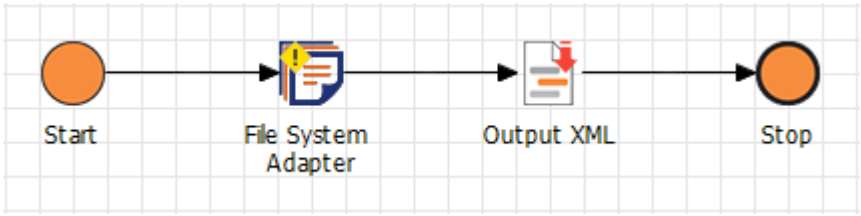
Weitere Informationen finden Sie im Abschnitt

- Operation: Write File
- Operation: Append File
- Operation: Read File
- Operation: Delete
- Operation: Copy
- Operation: Move
- Operation: Exists
- Operation: Create Folder
- Operation: List Folder

10.4.2.3 Operation: Read File

Die Operation `Read File` liest die Daten aus der im Parameter `path` festgelegten Datei.

10.4.2.3.1 Beispiele



Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Read File`, um die Daten aus der im Parameter `path` angegebenen Zieldatei zu lesen.

Parameter	Wert
`path`	C:\MyFolder\File.xml

Der Output des Adapters wird anschließend in eine XML-Datei im X4 Repository geschrieben.

Weitere Informationen finden Sie im Abschnitt

- Operation: Write File
- Operation: Append File
- Operation: Read File
- Operation: Delete
- Operation: Copy
- Operation: Move
- Operation: Exists
- Operation: Create Folder
- Operation: List Folder

10.4.2.4 Operation: Delete

Die Operation `Delete` löscht die im Parameter `path` festgelegte lokale Datei oder Ordner.

> ⚠ Die Daten werden unwiderruflich gelöscht.

10.4.2.4.1 Beispiele

10.4.2.4.1.1 Datei löschen

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Delete`, um die im Parameter `path` angegebene Zieldatei zu löschen.

Parameter	Wert
`path`	C:\MyFolder\File.xml

10.4.2.4.1.2 Ordner löschen

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Delete`, um den im Parameter `path` angegebenen Zielordner zu löschen.

Parameter	Wert
`path`	C:\MyFolder

> ✓ Weitere Informationen finden Sie im Abschnitt
>
> - Operation: Write File
> - Operation: Append File
> - Operation: Read File
> - Operation: Delete
> - Operation: Copy
> - Operation: Move
> - Operation: Exists
> - Operation: Create Folder
> - Operation: List Folder

10.4.2.5 Operation: Copy

Die Operation `Copy` kopiert die im Parameter `path` festgelegte Datei oder Ordner in den im Parameter `targetPath` festgelegten lokalen Ordner.

10.4.2.5.1 Beispiele

10.4.2.5.1.1 Ordner kopieren

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Copy`, um den Ordner `MyFolder` in das Verzeichnis `X4` zu kopieren:

Parameter	Wert
`path`	C:\MyFolder
`targetPath`	C:\X4\MyFolder

10.4.2.5.1.2 Datei kopieren

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Copy`, um die Datei `MyFile.xml` in das Verzeichnis `X4` zu kopieren:

Parameter	Wert
`path`	C:\MyFolder\MyFile.xml
`targetPath`	C:\X4\MyFolder\MyFile.xml

Weitere Informationen finden Sie im Abschnitt

- Operation: Write File
- Operation: Append File
- Operation: Read File
- Operation: Delete
- Operation: Copy
- Operation: Move
- Operation: Exists
- Operation: Create Folder
- Operation: List Folder

10.4.2.6 Operation: Move

Die Operation `Move` verschiebt die im Parameter `path` festgelegte Datei oder Ordner in den im Parameter `targetPath` festgelegten lokalen Ordner.

10.4.2.6.1 Beispiele

10.4.2.6.1.1 Ordner verschieben

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Move`, um den Ordner `MyFolder` in das Verzeichnis `X4` zu verschieben:

Parameter	Wert
`path`	C:\MyFolder
`targetPath`	C:\X4\MyFolder

10.4.2.6.1.2 Datei verschieben

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Move`, um die Datei `MyFile.xml` in das Verzeichnis `X4` zu verschieben:

Parameter	Wert
`path`	C:\MyFolder\MyFile.xml
`targetPath`	C:\X4\MyFolder\MyFile.xml

Weitere Informationen finden Sie im Abschnitt

- Operation: Write File
- Operation: Append File
- Operation: Read File
- Operation: Delete
- Operation: Copy
- Operation: Move
- Operation: Exists
- Operation: Create Folder
- Operation: List Folder

10.4.2.7 Operation: Exists

Die Operation `Exists` prüft, ob die im Parameter `path` festgelegten Dateien oder Ordner existieren. Gibt als Ergebnis eine XML mit Metadaten aus.

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Datei oder der Ordner existiert nicht.

10.4.2.7.1 Beispiele

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Exist`, um zu prüfen, ob die in Parameter `path` angegebene Zieldatei existiert.

Parameter	Wert
`path`	C:\MyFolder\File.xml

10.4.2.7.1.1 Output

Wenn die Datei nicht existiert:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>java.nio.file.NoSuchFileException: C:\MyFolder\File.xml</Error>
```

Wenn die Datei existiert:

```
<?xml version="1.0" encoding="UTF-8"?>
<Entry isDirectory="false" name="File.xml">
   <CreationDate>2021-06-18T08:59:20.910681Z</CreationDate>
   <LastModified>2021-06-18T08:22:59.284859Z</LastModified>
</Entry>
```

Wenn geprüft wird, ob ein Ordner existiert, wird das Attribut `isDirectory="true"` gesetzt.

Weitere Informationen finden Sie im Abschnitt

- Operation: Write File
- Operation: Append File
- Operation: Read File
- Operation: Delete
- Operation: Copy
- Operation: Move
- Operation: Exists
- Operation: Create Folder
- Operation: List Folder

10.4.2.8 Operation: Create Folder

Die Operation `Create Folder` erstellt den im Parameter `path` festgelegten Ordner.

10.4.2.8.1 Beispiele

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Create Folder`, um den im Parameter `path` angegebenen Order zu erstellen.

Parameter	Wert
`path`	C:\MyFolder

Weitere Informationen finden Sie im Abschnitt

- Operation: Write File
- Operation: Append File
- Operation: Read File
- Operation: Delete
- Operation: Copy
- Operation: Move
- Operation: Exists
- Operation: Create Folder
- Operation: List Folder

10.4.2.9 Operation: List Folder

Die Operation `List Folder` listet Namen und Metadaten von Dateien und Ordnern aus dem im Parameter `path` festgelegten Ordner auf. Gibt als Ergebnis eine XML mit Metadaten aus.

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Der Ordner existiert nicht.

10.4.2.9.1 Beispiele

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `List Folder`, um PDF-Dateien aus dem im Parameter `path` festgelegten Ordner aufzulisten.

Parameter	Wert
`path`	C:\MyFolder
`nameFilter`	*.pdf

10.4.2.9.1.1 Output

```
<?xml version="1.0" encoding="UTF-8"?>
<Entries>
   <Entry isDirectory="false" name="Document v1.pdf">
      <CreationDate>2021-06-18T09:40:39.1965Z</CreationDate>
      <LastModified>2021-06-18T09:20:49.137675Z</LastModified>
   </Entry>
   <Entry isDirectory="false" name="Document v2.pdf">
      <CreationDate>2021-06-18T09:40:07.105917Z</CreationDate>
      <LastModified>2021-06-18T09:20:49.137675Z</LastModified>
   </Entry>
   <Entry isDirectory="false" name="Document v3.pdf">
      <CreationDate>2021-06-18T09:40:18.941018Z</CreationDate>
      <LastModified>2021-06-18T09:20:49.137675Z</LastModified>
   </Entry>
</Entries>
```

Wenn ein Ordner gelistet wird, wird das Attribut `isDirectory="true"` gesetzt.

Weitere Informationen finden Sie im Abschnitt

- Operation: Write File
- Operation: Append File
- Operation: Read File
- Operation: Delete
- Operation: Copy
- Operation: Move
- Operation: Exists
- Operation: Create Folder
- Operation: List Folder

10.4.3 File System Bulk

`File System Bulk`: Vereinfacht die Nutzung von Datei-Operationen in Technical Processes. Mit ihm lassen sich einfach Aktionen wie das Auflisten, Verschieben und Kopieren von mehreren Dateien oder Verzeichnissen durchführen. Durch die Kompatibilität mit dem ZIP in Memory Compression (veraltet) Adapter lassen sich auch einfach multiple Dateien und Verzeichnisse (ent-)packen.

Der File System Bulk Adapter ist abgekündigt und wird in zukünftigen Versionen der X4 BPMS nicht mehr unterstützt. Bitte verwenden Sie stattdessen den File System Adapter.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • `Read`: Dateien auslesen • `Write`: Dateien und Verzeichnisse erzeugen/ersetzen • `Delete`: Dateien und Verzeichnisse löschen • `GetMetadata`: Metadaten von Dateien und Verzeichnissen auslesen • `Copy`: Dateien und Verzeichnisse kopieren • `Move`: Dateien und Verzeichnisse verschieben • `List`: Dateien und Verzeichnisse listen • `ListAndRead`: Dateien und Verzeichnisse listen und den Inhalt der Dateien gleich mit auslesen. (Kombiniert `List` mit `Read`) • `ListWithMetadata`: Dateien und Verzeichnisse listen und die Metadaten gleich mit auslesen. (Kombiniert `List` mit `GetMetadata`)

Die Input- und Output-Strukturen sind kompatibel mit denen des ZIP in Memory Compression (veraltet) Adapters.

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.filesystem.Bulk:* Hauptklasse (Standard)
baseDirectory	Basis-Verzeichnis für alle Operationen Alle Operationen können nur in diesem Verzeichnis und seinen Unterverzeichnissen arbeiten. **Mögliche Werte:** Absoluter Pfad zum Basis-Verzeichnis
subDirectory	Verzeichnis relativ zum `baseDirectory` als Basis für alle Operationen. **Mögliche Werte:** • *<leer>:* die Operationen starten im `baseDirectory` • `.`: die Operationen starten im `baseDirectory` • *a*: die Operationen starten in `<baseDirectory>/a` • `..` oder *a/../..*: Fehler, das `baseDirectory` kann auf diesem Weg nicht verlassen werden

limit	Begrenz die Anzahl der zurückgegebenen Treffer einer List-Operation. **Mögliche Werte:** • *<positive Zahl>:* maximale Anzahl der zurückgelieferten Treffer • *0*: Ergebnis ist eine leere Liste • *<negative Zahl>*: maximale Treffer-Anzahl ist unbegrenzt ⓘ Bei der Suche nach Dateien wird empfohlen, den Parameter `skipDirectories` auf *true* zu setzen, da sonst die gewünschten Treffer von Verzeichnissen verdrängt werden könnten.
maxLevel	Beschränkt die Rekursionstiefe für List-Operationen **Mögliche Werte:** • *<positive Zahl>*: maximale Tiefe, die eine List-Operation in Verzeichnisse hinabsteigt • *1*: nur der direkte Inhalt des Verzeichnisses, das durch `baseDirectory`/`subDirectory` angegeben ist • *0*: Ergebnis ist eine leere Liste • *<negative Zahl>*: die Rekursionstiefe ist unbegrenzt
filter	Regulärer Ausdruck, der auf den relativen Pfad eines Eintrags bei List-Operationen angewendet wird. Wenn der reguläre Ausdruck zutrifft, wird der Pfad in die Ergebnismenge der Liste übernommen.
skipDirectories	Definiert, ob eine List-Operation Verzeichnis-Elemente im Ergebnis ausgibt **Mögliche Werte:** • *true*: Verzeichnisse werden nicht in die Ergebnismenge von List-Operationen übernommen (nützlich in Verbindung mit `limit`) • *false*: Verzeichnisse werden in die Ergebnismenge von List-Operationen übernommen
overwrite	Definiert, ob bei den Operationen `Write`, `Move` oder `Copy` vorhandene Dateien überschrieben werden dürfen **Mögliche Werte:** • *true:* Bestehende Dateien dürfen bei `Write`, `Copy` und `Move` überschrieben werden. • *false:* Bestehende Dateien dürfen nicht überschrieben werden.

strict	Definiert, ob für `Write`-Operationen ein Elternverzeichnisse vorhanden sein muss und ob das Verzeichnis für die Operation `Delete` leer sein muss.
	
	Mögliche Werte:
	
	• *`true`*:
	  • Bei `Copy` und `Move` muss das Zielverzeichnis existieren, in welches hineinkopiert wird.
	  • Bei `Delete` müssen die Verzeichnisse leer sein.
	  • Bei `Write` muss das Zielverzeichnis existieren, in das hineingeschrieben wird.
	• *`false`*: Die Elternverzeichnisse werden für `Write`-Operationen angelegt. Zudem werden Verzeichnisse auch gelöscht, wenn sie noch nicht leer sind.

Statuswerte

1	Alle Einträge für die Operation konnten erfolgreich durchgeführt werden.
0	Nicht alle Einträge für die Operation konnten erfolgreich durchgeführt werden, d.h. das Ergebnis enthält `Error`-Elemente.
-1	Bei der Ausführung kam es zu anderen technischen Problemen, z.B. ungültiger Input.

Input

Input-Struktur für Read, Write, GetMetadata und Delete

```
<any>
 <!-- any mixture of file and directory elements -->
 <File name="relativePath">...Base64 content for Write...</File>
 <Directory name="relativePath" />
</any>
```

Input-Struktur für Copy und Move

```
<any>
 <!-- any mixture of file and directory elements -->
 <File name="relativePath" target="relativePath" />
 <Directory name="relativePath" target="relativePath" />
</any>
```

Output

```
<Data>
  <!-- any mixture of file, directory and error elements -->
  <File name="relativePath" [size="size of file in bytes"]
      [creationTime=""] [creationTimeAsMillis=""]
      [lastModifiedTime=""] [lastModifiedTimeAsMillis=""]
      [lastAccessTime=""] [lastAccessTimeAsMillis=""]>
    <!-- base64 conten, if operation Read or ListAndRead -->
    <!-- xml structures containing metadata, if operation GetMetadata or
ListWithMetadata -->
  </File>
  <Directory name="relativePath" [size="0"]
      [creationTime=""] [creationTimeAsMillis=""]
      [lastModifiedTime=""] [lastModifiedTimeAsMillis=""]
      [lastAccessTime=""] [lastAccessTimeAsMillis=""]>
    <!-- xml structures containing metadata, if operation GetMetadata or
ListWithMetadata -->
  </Directory>
  <Error name="relativePath" [target="relativePath"] exception="fully qualified
classname of exception"
      [size="size of file in bytes"]
      [creationTime=""] [creationTimeAsMillis=""]
      [lastModifiedTime=""] [lastModifiedTimeAsMillis=""]
      [lastAccessTime=""] [lastAccessTimeAsMillis=""]>
    <!-- base64 content if operation Write -->
  </Error>
</Data>
```

ⓘ Write, Delete, Copy und Move haben nur dann Error-Elemente als Ergebnis, wenn es tatsächlich Fehler gab.

10.4.3.1 Beispiele für List

10.4.3.1.1 **Voraussetzung**

Folgende Verzeichnisstruktur ist gegeben:

```
/
+ a/
| + index.html
| + logo.gif
| + content/
|   + toc.html
|   + page1.html
|   + page2.html
|   + images/
|     + pic1.png
|     + pic2.png
+ b/
  + some.html
  + other.html
  + files.html
```

10.4.3.1.2 Szenarien

10.4.3.1.2.1 Alles listen

Adapter-Konfiguration	Output
• baseDirectory = /a • subDirectory = • limit = -1 • maxLevel = -1 • filter = • skipDirectories = false	<Data> <File name="index.html" ... /> <File name="logo.gif" ... /> <Directory name="content" ... /> <File name="content/toc.html" ... /> <File name="content/page1.html" ... /> <File name="content/page2.html" ... /> <Directory name="content/images" ... /> <File name="content/images/pic1.png" ... /> <File name="content/images/pic2.png" ... /> </Data>

10.4.3.1.2.2 Die ersten 5 Einträge

Adapter-Konfiguration	Output
• baseDirectory = /a • subDirectory = • limit = 5 • maxLevel = -1 • filter = • skipDirectories = false	<Data> <File name="index.html" ... /> <File name="logo.gif" ... /> <Directory name="content" ... /> <File name="content/toc.html" ... /> <File name="content/page1.html" ... /> </Data>

10.4.3.1.2.3 Die ersten fünf Dateien

Adapter-Konfiguration	Output
• baseDirectory = /a • subDirectory = • limit = 5 • maxLevel = -1 • filter = • skipDirectories = true	<Data> <File name="index.html" ... /> <File name="logo.gif" ... /> <File name="content/toc.html" ... /> <File name="content/page1.html" ... /> <File name="content/page2.html" ... /> </Data>

10.4.3.1.2.4 Nur das aktuelle Verzeichnis

Adapter-Konfiguration	Output
• baseDirectory = /a • subDirectory = • limit = -1 • maxLevel = 1 • filter = • skipDirectories = false	<Data> <File name="index.html" ... /> <File name="logo.gif" ... /> <Directory name="content" ... /> </Data>

10.4.3.1.2.5 Alle Dateien und Verzeichnisse, die einen Filter erfüllen

Adapter-Konfiguration	Output
• baseDirectory = /a • subDirectory = • limit = -1 • maxLevel = -1 • filter = tent • skipDirectories = false	<Data> <Directory name="content" ... /> <File name="content/toc.html" ... /> <File name="content/page1.html" ... /> <File name="content/page2.html" ... /> <Directory name="content/images" ... /> <File name="content/images/pic1.png" ... /> <File name="content/images/pic2.png" ... /> </Data>

10.4.3.1.2.6 Alle Dateien, die einen Filter erfüllen (ohne Verzeichnisse)

Adapter-Konfiguration	Output
• baseDirectory = /a • subDirectory = • limit = -1 • maxLevel = -1 • filter = tent • skipDirectories = true	<Data> <File name="content/toc.html" ... /> <File name="content/page1.html" ... /> <File name="content/page2.html" ... /> <File name="content/images/pic1.png" ... /> <File name="content/images/pic2.png" ... /> </Data>

10.4.3.1.2.7 Alles unterhalb eines Unterverzeichnisses

Adapter-Konfiguration	Output
• baseDirectory = /a • subDirectory = content • limit = -1 • maxLevel = -1 • filter = • skipDirectories = false	<Data> <File name="toc.html" ... /> <File name="page1.html" ... /> <File name="page2.html" ... /> <Directory name="images" ... /> <File name="images/pic1.png" ... /> <File name="images/pic2.png" ... /> </Data>

10.4.3.1.2.8 Alles unterhalb des aktuellen Verzeichnisses und den direkten Kindverzeichnissen

Adapter-Konfiguration	Output
• baseDirectory = /a • subDirectory = • limit = -1 • maxLevel = 2 • filter = • skipDirectories = false	<Data> <File name="index.html" ... /> <File name="logo.gif" ... /> <Directory name="content" ... /> <File name="content/toc.html" ... /> <File name="content/page1.html" ... /> <File name="content/page2.html" ... /> <Directory name="content/images" ... /> </Data>

10.4.3.1.2.9 Das Startverzeichnis kann tief im Verzeichnisbaum liegen

Adapter-Konfiguration	Output
• baseDirectory = /a • subDirectory = content/images • limit = -1 • maxLevel = -1 • filter = • skipDirectories = false	<Data> <File name="pic1.png" ... /> <File name="pic2.png" ... /> </Data>

10.4.4 FTP

10.4.4.1 FTP Connector

`FTP Connector`: Unterstützt die Protokolle FTP (*File Transfer Protocol*), SFTP (*SSH File Transfer Protocol*) und FTPS (*File Transfer Protocol Secure* bzw. *FTP-SSL*) zum Senden, Empfangen, Löschen und Verschieben von einer Datei oder einem Ordner sowie zum Auflisten von Ordnerinhalten. Der Adapter unterstützt Proxy-Server vom Typ `HTTP`, `SOCKS4` und `SOCKS5`.

Ehemalige Bezeichnung bis Version 4.5.1: *File Transfer Adapter*

ⓘ Dieser Funktions-Adapter ersetzt den *FTP Transfer* sowie den *SFTP Communication* Adapter.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Dateitransfer-Operation mit den verarbeiteten Dateien oder Ordnern durchgeführt wird. **Mögliche Werte:** • `Exists`: Prüfen, ob die im Parameter `path` festgelegten Dateien oder Ordner existieren. Gibt als Ergebnis eine XML mit Metadaten aus. • `List`: Namen und Metadaten von Dateien und Ordnern aus dem im Parameter `path` festgelegten Ordner auflisten. Gibt als Ergebnis eine XML mit Metadaten aus. • ⓘ Der Inhalt der Dateien wird nicht gelistet. • `Send File`: Daten aus dem vorangegangenen Prozess-Schritt in die im Parameter `path` angegebene Zieldatei schreiben. • `Receive File`: Im Parameter `path` festgelegte Datei lesen. • `Append File`: Datei aus dem vorangegangenen Prozess-Schritt in die im Parameter `path` angegebenen Zieldatei anhängen. • `Move`: Datei oder Ordner auf dem FTP(S)/SFTP-Server von `path` nach `targetPath` verschieben. • `Delete File`: Im Parameter `path` festgelegte Dateien löschen. • `Create Folder`: Im Parameter `path` festgelegten Ordner auf dem FTP(S)/SFTP-Server erstellen. • `Delete Folder`: Im Parameter `path` festgelegten Ordner löschen.

Parameter

`host`	IP-Adresse oder Host-Name des FTP(S)/SFTP-Servers **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *`server.tld`*)
`port`	Port-Nummer des FTP(S)/SFTP-Servers **Mögliche Werte:** • Beliebige positive ganze Zahl • *22*: SFTP-Standardport (Standard)
`user`	Benutzername für die FTP(S)/SFTP-Verbindung
`password`	Passwort für die FTP(S)/SFTP-Verbindung
`path`	Pfad zu einem Ordner oder einer Datei auf dem FTP(S)/SFTP-Server **Mögliche Werte:** Beliebige Zeichenkette (z.B. */Folder/Subfolder* oder */Folder/Subfolder/File.xml*)

`targetPath`	Zusätzlicher Pfad zu einem Ordner oder einer Datei auf dem FTP(S)/SFTP-Server ⚠ Dieser Parameter wird nur bei der Operation `Move` verwendet. **Mögliche Werte:** Beliebige Zeichenkette (z.B. `/Folder/Subfolder` oder `/Folder/Subfolder/File.xml`)
`protocol`	Dateitransfer-Protokoll **Mögliche Werte:** • *`FTP`*: Operationen über das *File Transfer Protocol* ausführen (Standard) • *`FTPS`*: Operationen über das *File Transfer Protocol Secure* ausführen • *`SFTP`*: Operationen über das *SSH File Transfer Protocol* ausführen
`connectMode`	Verbindungsmodus **Mögliche Werte:** • *`ACTIVE:`* Aktiver Verbindungmodus • *`PASSIVE:`* Passiver Verbindungsmodus (standard)
`nameFilter`	Filtert den Output der Operation `List` nach Name **Mögliche Werte:** • Beliebige Zeichenkette (z.B. *`.xml`*) • Platzhalter (Wildcard) `*`
`addNewLine`	Fügt in der Zieldatei eine neue Zeile vor der anzuhängenden Datei ein. Kann nur mit der Operation `Append File` verwendet werden. **Mögliche Werte:** • `None`: (Standard) Fügt keine neue Zeile ein. • `Auto`: Fügt eine neue Zeile ein, wenn zuvor keine neue Zeile eingefügt wurde. Erkennt das Format einer bereits eingefügten neuen Zeile. • `Windows`: Fügt eine neue Zeile im Windows-Format ein. • `Unix`: Fügt eine neue Zeile im Unix-Format ein.

`hostValidationEnabled`	Host-Validierung aktivieren, um sicherzustellen, dass der angefragte SFTP-Server auch dem SFTP-Server entspricht, mit dem verbunden wird. ⚠ Der Parameter `hostValidationEnabled` ist nur für das SFTP-Protokoll relevant. **Mögliche Werte:** • *Ja*: Host-Validierung aktivieren (Flag setzen) • *Nein*: Host-Validierung deaktivieren (nicht empfohlen für Produktivsysteme) (Standard)
`hostValidationFile`	Pfad zur *Known Hosts*-Datei, die eine Liste von bestätigten Hosts und deren zugehörigen Public Keys enthält. **Mögliche Werte:** Beliebiger gültiger Pfad (z.B. *`C:\Data\known_hosts`*)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

✓ Weitere Informationen finden Sie im Abschnitt

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

10.4.4.1.1 Operation: List

Die Operation `List` listet Namen und Metadaten von Dateien und Ordnern aus dem im Parameter `path` festgelegten Ordner auf. Gibt als Ergebnis eine XML mit Metadaten aus.

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Der Ordner existiert nicht.

10.4.4.1.1.1 Beispiele

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `List`, um PDF-Dateien aus dem im Parameter `path` festgelegten Ordner aufzulisten.

Parameter	Wert
`path`	/MyFolder
`nameFilter`	*.pdf

Output

```
<?xml version="1.0" encoding="UTF-8"?>
<Entries>
   <Entry name="Document v1.pdf" isDirectory="false">
      <LastModified>2021-06-18T09:20:49.137675Z</LastModified>
   </Entry>
   <Entry name="Document v2.pdf" isDirectory="false">
      <LastModified>2021-06-18T09:20:49.137675Z</LastModified>
   </Entry>
   <Entry name="Document v3.pdf" isDirectory="false">
      <LastModified>2021-06-18T09:20:49.137675Z</LastModified>
   </Entry>
</Entries>
```

Wenn ein Ordner gelistet wird, wird das Attribut `isDirectory="true"` gesetzt.

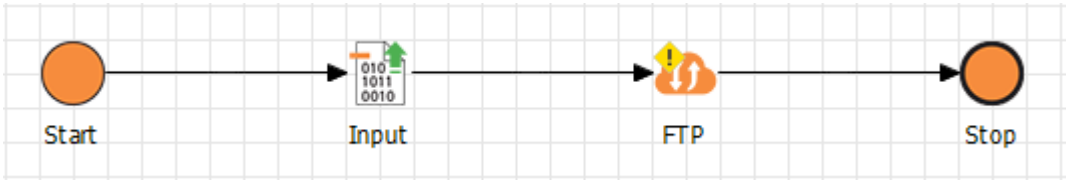
Weitere Informationen finden Sie im Abschnitt

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

10.4.4.1.2 Operation: Send File

Die Operation `Send File` schreibt Daten aus dem vorangegangenen Prozess-Schritt in die in Parameter `path` angegebene Zieldatei.

10.4.4.1.2.1 Beispiele



Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Send File`, um die Daten aus dem vorangegangenen Prozess-Schritt in die in Parameter `path` angegebene Zieldatei zu schreiben.

Parameter	Wert
`path`	/MyFolder/MyFile.xml

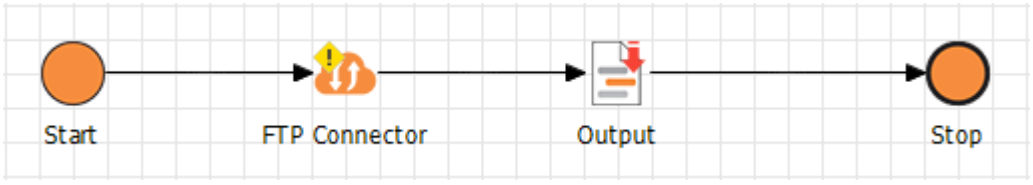
Weitere Informationen finden Sie im Abschnitt

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

10.4.4.1.3 Operation: Receive File

Die Operation `Receive File` liest die Daten aus der im Parameter `path` festgelegten Datei.

10.4.4.1.3.1 Beispiele



Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Receive File`, um die Daten aus der im Parameter `path` angegebenen Zieldatei zu lesen.

Parameter	Wert
`path`	/MyFolder/MyFile.xml

Der Output des Adapters wird anschließend in eine XML-Datei im X4 Repository geschrieben.

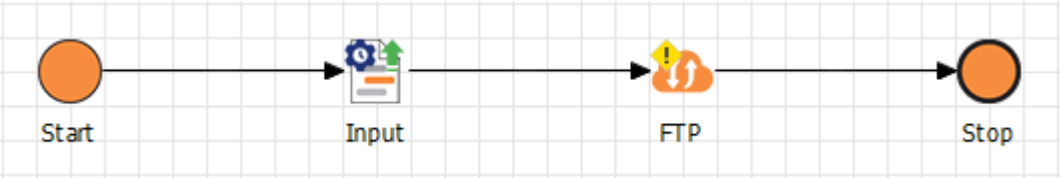
Weitere Informationen finden Sie im Abschnitt

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

10.4.4.1.4 Operation: Append File

Die Operation `Append File` hängt die Daten aus dem vorangegangenen Prozess-Schritt in die in Parameter `path` angegebene Zieldatei.

10.4.4.1.4.1 Beispiele



Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Append File`, um die Daten aus dem vorangegangenen Prozess-Schritt in die in Parameter `path` angegebene Zieldatei anzuhängen.

Parameter	Wert
`path`	/MyFolder/File.xml
`addNewline`	`Auto`

Zieldatei

```
<?xml version="1.0" encoding="UTF-8"?>
<Cars>
    <Car>
        <Brand>Audi</Brand>
        <Country>Germany</Country>
    </Car>
</Cars>
```

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Cars>
    <Car>
        <Brand>BMW</Brand>
        <Country>Germany</Country>
    </Car>
    <Car>
        <Brand>Hyundai</Brand>
        <Country>South Korea</Country>
    </Car>
</Cars>
```

Output

```
<?xml version="1.0" encoding="UTF-8"?>
<Cars>
    <Car>
        <Brand>Audi</Brand>
        <Country>Germany</Country>
    </Car>
</Cars>
<?xml version="1.0" encoding="UTF-8"?>
<Cars>
    <Car>
        <Brand>BMW</Brand>
        <Country>Germany</Country>
    </Car>
    <Car>
        <Brand>Hyundai</Brand>
        <Country>South Korea</Country>
    </Car>
</Cars>
```

Bei dem Output handelt es sich um eine nicht valide XML-Datei. Der Output muss im Anschluss mithilfe eines Mappings in eine valide XML-Datei transformiert werden.

Weitere Informationen finden Sie im Abschnitt

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

10.4.4.1.5 Operation: Move

Die Operation `Move` verschiebt die im Parameter `path` festgelegte Datei oder Ordner in den im Parameter `targetPath` festgelegten Ordner.

10.4.4.1.5.1 Beispiele

Ordner verschieben
Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Move`, um den Ordner `MyFolder` in den Ordner `X4` zu verschieben:

Parameter	Wert
`path`	/MyFolder
`targetPath`	/X4/MyFolder

Datei verschieben
Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Move`, um die Datei `MyFile.xml` in den Ordner `X4` zu verschieben:

Parameter	Wert
`path`	/MyFolder/MyFile.xml
`targetPath`	/X4/MyFile.xml

Weitere Informationen finden Sie im Abschnitt

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

10.4.4.1.6 Operation: Delete File

Die Operation `Delete File` löscht die im Parameter `path` festgelegte Datei.

Die Daten werden unwiderruflich gelöscht.

10.4.4.1.6.1 Beispiele

Datei löschen

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Delete File`, um die im Parameter `path` angegebene Zieldatei zu löschen.

Parameter	Wert
`path`	/MyFolder/MyFile.xml

Weitere Informationen finden Sie im Abschnitt

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

10.4.4.1.7 Operation: Create Folder

Die Operation `Create Folder` erstellt den im Parameter `path` festgelegten Ordner.

10.4.4.1.7.1 Beispiele

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Create Folder`, um den im Parameter `path` angegebenen Order zu erstellen.

Parameter	Wert
`path`	/MyFolder

Weitere Informationen finden Sie im Abschnitt

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

10.4.4.1.8 Operation: Delete Folder

Die Operation `Delete Folder` löscht den im Parameter `path` festgelegten Ordner.

⚠ Die Daten werden unwiderruflich gelöscht.

10.4.4.1.8.1 Beispiele

Ordner löschen
Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Delete Folder`, um den im Parameter `path` angegebenen Zielordner zu löschen.

Parameter	Wert
`path`	/MyFolder

✓ Weitere Informationen finden Sie im Abschnitt

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

10.4.4.1.9 Operation: Exists

Die Operation `Exists` prüft, ob die im Parameter `path` festgelegten Dateien oder Ordner existieren. Gibt als Ergebnis eine XML mit Metadaten aus.

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Datei oder der Ordner existiert nicht.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.4.4.1.9.1 Beispiele

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Exists`, um zu prüfen, ob die in Parameter `path` angegebene Zieldatei existiert.

Parameter	Wert
`path`	/file.xml

Output
Wenn die Datei existiert:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Entry name="file.xml" isDirectory="false">
    <LastModified>2022-03-07T12:33:23+01:00</LastModified>
</Entry>
```

Wenn geprüft wird, ob ein Ordner existiert, wird das Attribut `isDirectory="true"` gesetzt.

10.4.4.2 FTP Connector (Advanced)

`FTP Connector`: Unterstützt die Protokolle FTP (*File Transfer Protocol*), SFTP (*SSH File Transfer Protocol*) und FTPS (*File Transfer Protocol Secure* bzw. *FTP-SSL*) zum Senden, Empfangen, Löschen und Verschieben von einer oder mehreren Dateien und Ordnern sowie zum Auflisten von Ordnerinhalten. Der Adapter unterstützt Proxy-Server vom Typ `HTTP`, `SOCKS4` und `SOCKS5`.

Ehemalige Bezeichnung bis Version 4.5.1: *File Transfer Adapter*

Eigenschaften (Properties)

Operation (action)	Bestimmt, welche Dateitransfer-Operation mit den verarbeiteten Dateien oder Ordnern durchgeführt wird. **Mögliche Werte:** • `List`: Liste der Dateien des in Parameter `remotePath` angegebenen Ordners als XML-Dokument ausgeben • `Send`: Daten aus dem vorangegangen Prozess-Schritt in den in Parameter `remotePath` angegebenen Ordner speichern, dabei den in Parameter `documentName` festgelegten Dateinamen verwenden. Oder mehrere Dateien von Quellordner in Parameter `localPath` an `remotePath` versenden, indem die Parameter `directoryMode` und `multipleFiles` auf *`true`* gesetzt werden; in Parameter `filter` kann zudem ein Filterausdruck angegeben werden. • `Receive`: Datei in den Prozess einlesen, deren Pfad in Parameter `remotePath` und deren Dateiname in Parameter `documentName` festgelegt wurde. Oder mehrere Dateien empfangen, indem die Parameter `directoryMode` und `multipleFiles` auf *`true`* und `localPath` auf den lokalen Zielordner gesetzt werden; in Parameter `filter` kann zudem ein Filterausdruck angegeben werden. • `Delete`: In Parameter `documentName` festgelegte Datei oder Ordner auf dem FTP(S)/SFTP-Server löschen, deren Pfad in Parameter `remotePath` festgelegt wird. Um nicht-leere Ordner zu löschen, zusätzlich Parameter `recurse` aktivieren. • `Move`: Eine(n) in den Parametern `oldPath` mit `oldDocumentname` festgelegte Datei oder festgelegten Ordner innerhalb des FTP(S)/SFTP-Servers nach `remotePath` mit `documentname` verschieben • `Make directory`: Ordner auf dem FTP(S)/SFTP-Server erzeugen, dessen Name in Parameter `documentName` und dessen Pfad in Parameter `remotePath` festgelegt wird • `Delete directory`: Ordner auf dem FTP(S)/SFTP-Server löschen, dessen Name in Parameter `documentName` und dessen Pfad in Parameter `remotePath` festgelegt wird

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.transferadapter.TransferAdapter*: Hauptklasse (Standard)

`protocol`	Dateitransfer-Protokoll **Mögliche Werte:** • *`FTP`*: Operationen über das *File Transfer Protocol* ausführen (Standard) • *`FTPS`*: Operationen über das *File Transfer Protocol Secure* ausführen • *`SFTP`*: Operationen über das *SSH File Transfer Protocol* ausführen
`host`	IP-Adresse oder Host-Name des FTP(S)/SFTP-Servers **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *`server.tld`*)
`port`	Port-Nummer des FTP(S)/SFTP-Servers **Mögliche Werte:** • Beliebige positive ganze Zahl • *22*: SFTP-Standardport (Standard)
`connectMode`	Verbindungsmodus **Mögliche Werte:** • *`ACTIVE:`* Aktiver Verbindungmodus • *`PASSIVE:`* Passiver Verbindungsmodus (standard)
`controlEncoding`	Verwendete Kodierung für den Kommando-Kanal. Hiermit wird beeinflusst wie Umlaute in Dateinamen übertragen werden. Dieser Wert ist abhängig vom FTP-Server. **Mögliche Werte:** UTF-8 (Standard) Ist kein `controlEncoding` gesetzt oder das `controlEncoding` entspricht dem Leerstring, wird die Verbindung nicht modifiziert und das `controlEncoding` ist undefiniert. Sonst wird der Wert als `controlEncoding` auf die Verbindung gesetzt.
`login`	Benutzername für die FTP(S)/SFTP-Verbindung
`password`	Passwort für die FTP(S)/SFTP-Verbindung; Wenn Parameter `privateKeyFile` gesetzt ist: Passwort des *Private Keys*
`accountInfo`	Zusätzliche Informationen zum Konto, siehe auch Dokumentation der FTP-Bibliothek

`loginCommandsUrl`	URL zu der im XStore liegenden XML-Datei, welche die Login-Commands enthält **Mögliche Werte:** Beliebige XStore-URL
`remotePath`	Pfad auf dem FTP(S)/SFTP-Server, dessen Zielordner per FTP(S)/SFTP manipuliert werden soll **Mögliche Werte:** Beliebige Zeichenkette (z.B. *`Folder/Subfolder`*)
`documentname`	Name des Ordners oder der Datei, der/die verarbeitet wird. Wenn `directoryMode=true` gesetzt wurde, wird im Parameter der Ordnername erwartet. **Mögliche Werte:** Beliebige Zeichenkette (z.B. *`Data.xml`*)
`oldPath`	Pfad zum Ordner oder zur Datei auf dem FTP(S)/SFTP-Server, der/die verschoben werden soll (nur bei Operation `Move`) **Mögliche Werte:** Beliebige Zeichenkette (z.B. *`Folder/Subfolder`*)
`oldDocumentname`	Name der Datei, die verschobenen werden soll (nur bei Operation `Move`) **Mögliche Werte:** Beliebige Zeichenkette (z.B. *`Data.xml`*)
`recurse`	Pfad rekursiv löschen, d.h. inklusive aller Unterordner und Dateien (nur bei Operation `Delete`) **Mögliche Werte:** • `True`: Alle beinhalteten Ordner (und Unterordner) und Dateien löschen • `False`: Nur Ordner und Dateien im aktuellen Pfad löschen (Standard)
`directoryMode`	Ordner-Modus aktivieren, um den Inhalt von Ordnern zu verarbeiten; falls Parameter `multipleFiles` aktiviert ist, werden alle Elemente verarbeitet, die dem Filterkriterium in Parameter filter entsprechen **Mögliche Werte:** • `true`: Ordner verarbeiten • `false`: Nur einzelne Dateien verarbeiten (Standard)

`localPath`	Lokaler Ordner zum Senden oder Empfangen von mehreren Dateien (nur wenn die Parameter `directoryMode` und `multipleFiles` aktiviert sind) **Mögliche Werte:** Beliebige gültige Pfadangabe (z.B. *`C:\Directory`*)
`multipleFiles`	Alle Dateien verarbeiten, die dem Filterkriterium in Parameter `filter` entsprechen **Mögliche Werte:** • `true`: Alle zutreffenden Dateien verarbeiten • `false`: Nur die Datei verarbeiten, die in Parameter `documentName` angegeben wurde, oder wenn Parameter `directoryMode` aktiviert ist, die erste Datei (falls das Filterkriterium auf mehrere Dateien zutrifft) (Standard)
`filter`	Filterausdruck, der die Platzhalter *und *?* enthalten kann, um die Menge der verarbeiteten Dateien einzuschränken. Dies gilt für die Operation `list` und die Operationen `receive` und `send` u nd nur, wenn die Parameter `multipleFiles` und `directoryMode` aktiviert sind. **Mögliche Werte:** Beliebige Zeichenkette (z.B. *`*.xml`*)
`deleteAfterReceive`	Datei oder Ordner nach dem Empfangen vom FTP(S)/SFTP-Server löschen (nur bei Operation `Receive`) **Mögliche Werte:** • `true`: Ursprüngliche Datei oder Ordner nach dem Empfangen auf dem FTP(S)/SFTP-Server löschen • `false`: Ursprüngliche Datei oder Ordner behalten (Standard)
`deleteAfterSend`	Datei oder Ordner nach dem Senden zum FTP(S)/SFTP-Server löschen (nur bei Operation `Send`) **Mögliche Werte:** • `true`: Ursprüngliche Datei oder Ordner nach dem Senden zum FTP(S)/SFTP-Server löschen • `false`: Ursprüngliche Datei oder Ordner behalten (Standard)

`append`	Gesendete Daten an die Zieldatei anhängen, wenn bereits eine gleichnamige Zieldatei besteht (nur bei Operation `Send`) **Mögliche Werte:** • `true`: Daten in der Zieldatei (falls vorhanden) anhängen • `false`: Zieldatei (falls vorhanden) überschreiben (Standard)
`transferMode`	Übertragungsart **Mögliche Werte:** • `BINARY`: Als Binärdatei übertragen (Standard) • `TEXT`: Als Text übertragen
`maxTransferRate`	Maximale Übertragungsrate in Bytes/Sekunde **Mögliche Werte:** • Beliebige positive ganze Zahl • `0`: keine Begrenzung (Standard)
`sleepEnabled`	Ruhemodus (*Sleep Mode*) aktivieren (wenn Parameter `multipleFiles` aktiviert ist) **Mögliche Werte:** • `true`: Ruhemodus aktivieren (ermöglicht die Definition einer Ruhedauer in Parameter `sleepTime`) • `false`: Ruhemodus deaktivieren (Standard)
`sleepTime`	Wartezeit in Sekunden zwischen der Übertragung der Dateien (wenn Parameter `multipleFiles` und `sleepEnabled` aktiviert sind) **Mögliche Werte:** • Beliebige positive ganze Zahl (Standard) • `0`: Keine Ruhezeit verwenden
`timeOut`	Zeitbeschränkung für den TCP-Verbindungsaufbau und die Übertragung in Millisekunden **Mögliche Werte:** • Beliebige positive ganze Zahl (z.B. *`50000`*) • `0`: Keine Zeitbeschränkung (Standard)

`hostValidationEnabled`	Host-Validierung aktivieren, um sicherzustellen, dass der angefragte FTP(S)/SFTP-Server auch dem FTP(S)/SFTP-Server entspricht, mit dem verbunden wird. Idealerweise nur für Testzwecke deaktivieren! **Mögliche Werte:** • `true`: Host-Validierung aktivieren (Flag setzen) ℹ Ist die Host-Validierung aktiviert, muss der Parameter `hostsFileName` gesetzt sein. • `false`: Host-Validierung deaktivieren (nicht empfohlen für Produktivsysteme) (Standard)
`implicitFTPS`	Implicit FTPS aktivieren **Mögliche Werte:** • `true`: Implicit FTPS aktivieren (Flag setzen) • `false`: Implicit FTPS deaktivieren (Standard)
`securityMechanism`	Sicherheitsmechanismus **Mögliche Werte:** • `Explicit_TLS` • `Explicit_SSL` • `Explicit_TLS_C` • `Implicit_PRIVATE` • `Implicit_CLEAR`
`privateKeyFile`	Pfad zu einer *Private Key*-Datei (nicht: *Java Keystore*). Wenn Sie sich auf dem FTP(S)/SFTP-Server mit einem *Private Key* authentifizieren möchten, dann dient der Parameter password als Passwort für Ihre *Private Key* -Datei. **Mögliche Werte:** Beliebiger gültiger Pfad (z.B. *`C:\Data\X4Keys\id_rsa`*)

`hostsFileName`	Pfad zur *Known Hosts*-Datei, die eine Liste von bestätigten Hosts und deren zugehörigen Public Keys enthält
	
	Mögliche Werte: Beliebiger gültiger Pfad (z.B. *`C:\Data\known_hosts`*)
	
	ℹ Die *Known Hosts*-Datei muss wie folgt aufgebaut sein:
	
	`host,IP Verschlüsselung Host_PublicKey`
	`ftp.softproject.de,217.160.223.49 ssh-rsa AAAAB3NzaC1yc2EAAAA...`

`useProxy`	Proxy-Server für den Verbindungsaufbau verwenden **Mögliche Werte:** • `true`: Proxy-Server verwenden • `false`: Keinen Proxy-Server verwenden (Standard)
`proxyType`	Proxy-Server-Typ **Mögliche Werte:** • `HTTP`: HTTP-Proxy verwenden (Standard) • `SOCKS4`: Proxy auf Basis des *SOCKS 4*-Protokolls verwenden • `SOCKS5`: Proxy auf Basis des *SOCKS 5*-Protokolls verwenden
`proxyAddress`	IP-Adresse oder Host-Name des Proxy-Servers **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *`server.tld`*)
`proxyPublicAddress`	Öffentliche IP-Adresse oder Host-Name des Proxy-Servers, wenn der Proxy keine Adresse zurückgibt (erforderlich z.B. bei aktiven FTP-Übertragungen, wenn der FTP(S)/SFTP-Server sich erneut mit dem Proxy verbinden muss) **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *`server.tld`*)
`proxyPort`	Port-Nummer des Proxy-Servers **Mögliche Werte:** Beliebige positive ganze Zahl
`proxyUsername`	Benutzername für den Proxy-Server
`proxyPassword`	Passwort für den Proxy-Server

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Im Datei-Modus (wenn Parameter `directoryMode` deaktiviert ist), wird für die Operation `Send` ein Input-Dokument erwartet, das versendet wird.

Die unterschiedlichen Login-Operationen haben eine Reihenfolge. Wenn die jeweiligen Parameter gefüllt sind, gilt:

1. Login-Commands werden verwendet
2. User/Passwort mit `AccountInfo` wird verwendet
3. User/Passwort wird verwendet

Definition der LoginCommands-XML:

```
<loginCommands>
    <loginCommand>cmd1</loginCommand>
    <loginCommand>cmd2</loginCommand>
</loginCommands>
```

Ausgabe

Im Datei-Modus (wenn Parameter `directoryMode` deaktiviert ist), wird für die Operation `Receive` das empfangene Dokument ausgegeben.

Für Operation `List` wird ein XML-Dokument mit der Liste der Ordner und Dateien ausgegeben.

10.4.5 iCalendar

`iCalendar`: Erzeugt aus einem adapterspezifischen XML-Dokument ein *iCalendar*-Ereignis gemäß RFC 5545 oder konvertiert ein *iCalendar*-Textdokument in ein XML-Dokument. Zudem kann ein *POP/SMTP*-Adapter dynamisch parametriert werden, sodass im Anhang einer E-Mail ein *iCalendar*-Ereignis (z. B. eine Termineinladung) versendet wird.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`iCal To XML`*: *iCal*-Textdokument in XML konvertieren • *`XML To iCal`*: Adapterspezifische XML-Struktur in ein *iCalendar*-Textdokument konvertieren • *`Create iXmap for E-Mail Adapter`*: *iXmap*-XML-Struktur mit Parameterwerten und dem *iCal*-Ereignis als E-Mail-Body ausgeben, die zum dynamischen Parametrieren eines `POP/SMTP-Adapters` dient; siehe *X4 Entwickler-Referenz – Dynamische Prozesse über iXmap.* ⚠ Diese Operation wird mit dem nächsten Release abgekündigt.

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.icalendar.ICalendarAdapter*: Hauptklasse (Standard)
attachmentName	Dateiname des E-Mail-Anhangs mit dem *iCal*-Kalenderelement (für Operation `Create iXmap for E-Mail Adapter`) **Mögliche Werte:** Beliebiger gültiger Dateiname

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Beispiel

Im Demosystem *X4 Experience* finden Sie den Beispielprozess `iCalendarAdapter.wrf` zur Verwendung des Adapters.

Abhängigkeiten

Dieser Adapter folgende Bibliotheken:

- `ical4j-1.0.jar`
- `backport-util-concurrent-3.1.jar`
- `icalendar.jar`

10.4.5.1 Operation: iCal to XML

Mit Operation `iCal To XML` lässt sich ein gültiges *iCal*-Textdokument in XML konvertieren.

Input

Für Operation `iCal To XML` erwartet der Adapter ein beliebiges gültiges *iCal*-Textdokument.

Ausgabe

Für Operation `iCal To XML` gibt der Adapter ein XML-Dokument aus, das sämtliche Termin-Informationen enthält (siehe Beschreibung der Input-Struktur bei Operation `XML To iCal`).

10.4.5.2 Operation: XML to iCal

Mit Operation `XML To iCal` lässt sich eine adapterspezifische XML-Struktur in ein *iCalendar*-Textdokument konvertieren.

Input

Der Adapter erwartet ein Input-XML-Dokument gemäß folgendem Muster:

```
<iCalendarData>
    <!-- E-Mail-Metadaten: Nur relevant, wenn der Adapter ein iXmap
         für den POP/SMTP Adapter erzeugen soll -->
   <mail>
      <from><!--E-Mail-Adresse des Absenders--></from>
      <to><!--E-Mail-Adressen der Empfänger durch Semikolons (";") getrennt--></to>
      <cc><!--E-Mail-Adressen der CC-Empfänger durch Semikolons getrennt--></cc>
      <bcc><!--E-Mail-Adressen der BCC-Empfänger durch Semikolons getrennt--></bcc>
      <subject><!--Betreff--></subject>
   </mail>
   <prodId><!--Programm-ID, das die iCalendar-Daten erzeugt hat--></prodId>

   <!-- Veröffentlichungsart gemäß RFC 2446 (S.73)
        z. B. "REQUEST": Besprechungsanfrage;
             "PUBLISH": rein informative Terminveröffentlichung -->
   <method><!--Veröffentlichungsart--></method>
   <timezoneId><!--Beliebige Zeitzonen-ID von ical4j, z. B. Europe/Berlin--></
timezoneId>

   <!-- Ereignis: siehe RFC 2445 S. 51 -->
   <vevent>
      <uid><!--Global eindeutige ID für die iCalendar-Komponente (ggf. automatisch
erzeugt)--></uid>
      <organizer>
         <name><!--Name des Organisators--></name>
         <mail><!--E-Mail-Adresse des Organisators--></mail>

        <!-- zusätzliche Parameter je nach iCalendar-Eigenschaft
            (nur für Experten) -->
         <parameters>
            <parameter>
               <name><!--Parametername--></name>
               <value><!--Parameterwert--></value>
            </parameter>
         </parameters>
      </organizer>

       <!-- Teilnehmer: (von MS Outlook nur teilweise unterstützt) -->
       <!-- Outlook verwendet stattdessen die Empfängerdaten der E-Mail -->
      <attendees>
         <attendee>
            <name><!--Name des Teilnehmers--></name>
            <mail><!--E-Mail des Teilnehmers--></mail>

             <!-- Teilnehmerinformationen; siehe RFC 2445 S. 28 -->
            <role><!--REQ-PARTICIPANT, OPT-PARTICIPANT oder NON-PARTICIPANT--></role>
            <partStat><!--Teilnahmestatus (bei Antwort): ACCEPTED, DECLINED ...--></
partStat>
            <askForResponse><!--Antwort anfordern: true oder false--></
askForResponse>

            <!-- zusätzliche Parameter je nach iCalendar-Eigenschaft
                (nur für Experten) -->
            <parameters>
```

```
               <name><!--Parametername--></name>
               <value><!--Parameterwert--></value>
            </parameters>
         </attendee>
      </attendees>

      <startTime timezoneId="Europe/London"><!--Startzeitpunkt (DATE oder DATE-
TIME)-->
      </startTime>
      <endTime><!--Endzeitpunkt des Termins (DATE oder DATE-TIME)--></endTime>

      <!-- Wenn ein Termin geändert wird: 0-n; wird bei
           jeder Änderung um 1 erhöht -->
      <sequence><!--Sequenz: 0-n--></sequence>
      <priority><!--Priorität von 1-9 (1=höchste Priorität, 0=keine Priorität)--></
priority>
      <summary><!--Titel des Termins--></summary>
      <description><!--Beschreibungstext des Termins--></description>
      <location><!--Ort--></location>
      <status><!--Status des Termins: CONFIRMED, TENTATIVE ...--></status>

      <!-- Hinweis: MS Outlook kann Termine mit gesetzten
           Elementen "rrule" und "exrule" nicht verarbeiten! -->
      <rrule><!--Wiederholungsregel--></rrule>
      <exrule><!--Ausnahmen der Wiederholungsregel--></exrule>

      <!-- siehe RFC 2445 S.66 -->
      <!-- Löst zu einem bestimmten Zeitpunkt eine Aktion aus;
           kann nur in "vevent" und "vtodo2" verwendet werden -->
      <valarm>

         <!-- siehe RFC 2445 S.70 -->
         <trigger>
            <value><!--Zeitangabe, deren Typ von Element "type" abhängt--></value>
            <type><!--Zeitangabentyp: DURATION (Standard), DATE-TIME oder DATE--></
type>

            <!-- wenn dieses Element "true" enthält und als Typ "DURATION"
                 gewählt ist, dann wird das Element "value" relativ zum
                 Endzeitpunkt der übergeordneten Komponente betrachtet,
                 andernfalls relativ zum Startzeitpunkt -->
            <relativeToEnd>Relative Angabe: true oder false--></relativeToEnd>
         </trigger>

         <!-- siehe RFC 2445 S.69 -->
         <action><!--DISPLAY--></action>
         <repeat><!--Anzahl an Wiederholungen des Alarms--></repeat>

         <!-- Zeitabstand zwischen den Alarmen, wenn "repeat" größer 0
              ist (hier 30 Sekunden) -->
         <duration><!--PT30S--></duration>
         <summary><!--Titel der Alarmmeldung--></summary>
         <description><!--Beschreibung der Alarmmeldung--></description>

         <!-- Optional: Definition von "attendees" und "otherProperties"
```

```
            (Struktur wie in "vevent") -->
      </valarm>

      <categories>
         <!-- 1-n Kategorien, zu denen der Termin gehören soll -->
         <category><!--Kategoriename--></category>
      </categories>

      <!-- Sonstige Properties (nur Properties, die für
           diese Komponente definiert sind) -->
      <otherProperties>
         <property>
            <name><!--Eigenschaftsname--></name>
            <value><!--Eigenschaftswert--></value>
            <parameters>
               <parameter>
                  <name><!--Parametername--></name>
                  <value><!--Parameterwert--></value>
               </parameter>
            </parameters>
         </property>
      </otherProperties>
   </vevent>

   <!-- Aufgabe: siehe RFC 2445 S.54 -->
   <vtodo>
      <due timezoneId="US-Eastern"><!--Endzeitpunkt im Format YYYYMMDDThhmmss--></
due>
      <duration><!--Dauer - nicht gemeinsam mit Element "due" verwenden--></duration>

      <!-- Hier können zusätzlich alle Elemente aus "vevent" außer "endTime"
           verwendet werden. -->
   </vtodo>

   <!-- Journal: siehe RFC 2445 S.55 -->
   <vjournal>
      <!-- Alle Elemente aus "vevent" außer: "sequence", "priority",
           "location", "endTime" und "valarm"  -->
   </vjournal>

   <!-- Belegt/Frei: siehe RFC 2445 S.57 -->
   <vfree_busy>
      <uid>(siehe Element vevent)</uid>
      <organizer><!--siehe Element "vevent"--></organizer>
      <attendees><!--siehe Element "vevent"--></attendees>
      <startTime><!--siehe Element "vevent"--></startTime>
      <endTime><!--siehe Element "vevent"--></endTime>
      <duration><!--Dauer - nicht gemeinsam mit Element "endTime" verwenden--></
duration>
      <comment><!--Kommentartext--></comment>
      <otherProperties><!--siehe Element "vevent"--></otherProperties>
   </vfree_busy>

   <otherProperties><!--siehe Element "vevent"--></otherProperties>
</iCalendarData>
```

Ausgabe

Für Operation `XML To iCal` gibt der Adapter ein entsprechendes *iCalendar*-Textdokument aus.

10.4.5.3 Operation: Create iXmap for E-Mail Adapter

⚠ Diese Operation wird mit dem nächsten Release abgekündigt.

Mit Operation `Create iXmap for E-Mail Adapter` lässt sich eine *iXmap*-XML-Struktur mit Parameterwerten und dem *iCal*-Ereignis als E-Mail-Body ausgeben, die zum dynamischen Parametrieren eines *POP/SMTP*-Adapters dient; siehe *X4 Developer Guide – Dynamische Prozesse über iXmap*.

Input

Für Operation `Create iXmap for E-Mail Adapter` erwartet der Adapter die gleiche XML-Struktur wie bei Operation `XML To iCal`.

Zusätzlich kann jedoch das Wurzel-Element als erstes Kindelement ein Element `<mail>` enthalten, in dem für jeden Parameter, der beim *POP/SMTP*-Adapter dynamisch gesetzt werden soll, ein gleichnamiges Element inklusive Wert definiert werden kann:

```
<iCalendarData>
   <mail>
      <!-- für jeden zu setzenden Parameter -->
      <Parametername 1><!--Wert 1--></Parametername 1>
      <Parametername n><!--Wert n--></Parametername n>
   </mail>
   <!-- Termindaten -->
</iCalendarData>
```

Ausgabe

Für Operation `Create iXmap for E-Mail Adapter` gibt der Adapter ein *iXmap*-XML-Dokument aus, das zum dynamischen Parametrieren eines *POP/SMTP*-Adapters dient; siehe *X4 Developer Guide – Dynamische Prozesse über iXmap*.

Damit die Prozess-Engine die *iXmap*-Struktur jedoch als solche erkennt, muss dieses XML-Dokument erneut via XSL transformiert werden (Identitätstransformation genügt). Ein darauf folgender *POP/SMTP*-Adapter wird dann dynamisch mit den im Input-XML-Dokument angegebenen Werten parametriert.

```
<iXmap>
   <Return>
      <!-- für jeden gesetzten Parameter -->
      <Parametername 1><!--Wert 1--></Parametername 1>
      <Parametername n><!--Wert n--></Parametername n>
   </Return>
</iXmap>
```

10.4.6 Ping

`Ping`: Pingt eine IP-Adresse über das *Internet Control Message Protocol* an. Dabei sendet der Adapter *ICMP ECHO REQUEST*s oder versucht alternativ, eine TCP-Verbindung auf Port 7 aufzubauen. Wenn über vorausgehende Requests keine Verbindung aufgebaut werden kann und der Port auf einen anderen Wert als 7 gesetzt ist, wird ein weiterer TCP-Request an den angegebenen Port gesendet. Der Adapter gibt ein XML-Dokument mit dem Ergebnis des Echos (*`true`* oder *`false`*) aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt. **Mögliche Werte:** • *`Ping`*: Ping ausführen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.icmpping.ICMPPing`*: Hauptklasse (Standard)
iPAddress	IP-Adresse, an die der Ping gesendet wird **Mögliche Werte:** • Beliebige IP-Adresse (z. B. *`127.0.0.1`*) • *`0.0.0.0`*: Leere IP-Adresse (Standard)
timeout	Zeitbeschränkung für den Verbindungsaufbau in Millisekunden **Mögliche Werte:** • Beliebige positive ganze Zahl • *`0`*: Keine Zeitbeschränkung (Standard)

port	Port für einen erneuten Verbindungsversuch, wenn der erste ICMP-Request fehlschlägt **Mögliche Werte:** • Beliebige positive ganze Zahl

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.4.7 SCP

`SCP`: Sendet und empfängt einzelne Dateien über SCP (*Secure CoPy*). Der Adapter unterstützt Proxy-Server vom Typ HTTP, SOCKS4 und SOCKS5.

Ehemalige Bezeichnung bis Version 4.5.1: *SCP Transfer Adapter*

Eigenschaften (Properties)

Operation (action)	Bestimmt, welche Dateitransfer-Operation mit den verarbeiteten Dateien oder Ordnern durchgeführt wird. **Mögliche Werte:** • *`Send`*: Eine in Parameter `documentname` angegebene Datei, die sich in einem in Parameter `localPath` festgelegten lokalen Ordner befindet, an einen entfernten Ordner versenden, der in Parameter `remotePath` angeben wurde. • *`Receive`*: Datei empfangen, deren Pfad in Parameter `remotePath` und deren Dateiname in Parameter `documentName` festgelegt wurde und in einem lokalen Ordner speichern, der in Parameter `localPath` angegeben wurde.

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.scpadapter.SCPAdapter`* : Hauptklasse (Standard)

host	IP-Adresse oder Host-Name des entfernten Rechners **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *`server.tld`*)
port	Port-Nummer des entfernten Rechners **Mögliche Werte:** • Beliebige positive ganze Zahl • *`22`*: SCP-Standardport (Standard)
login	Benutzername für die SCP-Verbindung
password	Passwort für die SCP-Verbindung; Wenn Parameter `privateKeyFile` gesetzt ist: Passwort des *Private Keys*
remotePath	Pfad auf dem entfernten Rechners, mit dem eine Datei ausgetauscht werden soll **Mögliche Werte:** Beliebige Zeichenkette (z.B. *`Folder/Subfolder`*)
documentname	Name der Datei, die gesendet oder empfangen werden soll **Mögliche Werte:** Beliebige Zeichenkette (z.B. *`Data.xml`*)
localPath	Lokaler Ordner zum Senden und Empfangen einer Datei **Mögliche Werte:** Beliebige gültige Pfadangabe (z.B. *`C:\Directory`*)
timeOut	Zeitbeschränkung für den TCP-Verbindungsaufbau und die Übertragung in Millisekunden **Mögliche Werte:** • Beliebige positive ganze Zahl (z.B. *`50000`*) • *`0`*: Keine Zeitbeschränkung (Standard)

hostValidationEnabled	Host-Validierung aktivieren, um sicherzustellen, dass der angefragte entfernte Rechner auch dem Rechner entspricht, mit dem verbunden wird. Idealerweise nur für Testzwecke deaktivieren! **Mögliche Werte:** • *Ja*: Host-Validierung aktivieren (Flag setzen) • *Nein*: Host-Validierung deaktivieren (nicht empfohlen für Produktivsysteme) (Standard)
privateKeyFile	Pfad zu einer *Private Key* -Datei (nicht: *Java Keystore*). Wenn Sie sich auf dem entfernten Rechner mit einem *Private Key* authentifizieren möchten, dann dient der Parameter `password` als Passwort für Ihre *Private Key*-Datei. **Mögliche Werte:** Beliebiger gültiger Pfad (z.B. *C:\Data\X4Keys\id_rsa*)
hostsFileName	Pfad zur *Known Hosts*-Datei, die eine Liste von bestätigten Hosts und deren zugehörigen Public Keys enthält **Mögliche Werte:** Beliebiger gültiger Pfad (z.B. *C:\Data\known_hosts*)
useProxy	Proxy-Server für den Verbindungsaufbau verwenden **Mögliche Werte:** • *Ja*: Proxy-Server verwenden • *Nein*: Keinen Proxy-Server verwenden (Standard)
proxyType	Proxy-Server-Typ **Mögliche Werte:** • *HTTP*: HTTP-Proxy verwenden (Standard) • *SOCKS4*: Proxy auf Basis des *SOCKS 4* -Protokolls verwenden • *SOCKS5*: Proxy auf Basis des *SOCKS 5* -Protokolls verwenden
proxyAddress	IP-Adresse oder Host-Name des Proxy-Servers **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *server.tld*)

proxyPublicAddress	Öffentliche IP-Adresse oder Host-Name des Proxy-Servers, wenn der Proxy keine Adresse zurückgibt **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *server.tld*)
proxyPort	Port-Nummer des Proxy-Servers **Mögliche Werte:** Beliebige positive ganze Zahl
proxyUsername	Benutzername für den Proxy-Server
proxyPassword	Passwort für den Proxy-Server

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Der Adapter erwartet kein bestimmtes Input-Dokument. Ein Input-Dokument wird unverändert wieder ausgegeben.

Ausgabe

Ein Input-Dokument wird unverändert wieder ausgegeben.

10.4.8 TCP/IP Connector/Listener

10.4.8.1 TCP/IP Communication

? Unknown Attachment `TCP/IP Communication`: Führt synchrone oder asynchrone TCP-Socket-Aufrufe aus, öffnet dabei einen Socket und sendet die Input-Daten an den angegeben Host.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** `Approach`: TCP-Socket aufrufen (Standard)

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.socketapproach.SocketApproach*: Hauptklasse (Standard)

iPAddress	IP-Adresse des Host-Rechners, an den Daten gesendet werden sollen **Mögliche Werte:** • Beliebige IP-Adresse (z.B. *127.0.0.1*) • *0.0.0.0*: Leere IP-Adresse (Standard)
port	Port des Host-Rechners **Mögliche Werte:** • Beliebige ganze positive Zahl (z.B. *8080*) • *0*: Port 0 (Standard)
timeout	Zeitbeschränkung für den Verbindungsaufbau in Millisekunden **Mögliche Werte:** • Beliebige positive ganze Zahl • *0*: Keine Zeitbeschränkung (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.4.9 Telnet Connector

`Telnet Connector`: Stellt eine Verbindung zu Servern über das Teletype Network Protocol her.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** `Execute`: Führt die Befehle aus, die über ein Input-XML zur Verfügung gestellt wurden

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.telnet.TelnetAdapter*: Hauptklasse (Standard)

`host`	*Erforderlicher Parameter*: Host des Servers zu dem die Verbindung hergestellt werden soll **Mögliche Werte:** • `IP`: Beliebige gültige Host-IP, z. B. *`1.1.1.1`* • `String`: Beliebiger gültiger Host-Name, z. B. *`myServer.com`*
`port`	Port des Servers zu dem die Verbindung hergestellt werden soll **Mögliche Werte:** • Beliebige ganze Zahl • `23` (Standard): Standard-Port für Telnet-Verbindungen
`user`	Benutzername für die Telnet-Verbindung **Mögliche Werte:** Zeichenkette mit dem Benutzernamen, z. B. *`MyUser`*
`password`	Passwort für die Telnet-Verbindung **Mögliche Werte:** Zeichenkette mit dem Passwort, z. B. *`MyPassword`*
`timeout`	Verbindungs-Timeout in Millisekunden **Mögliche Werte:** • Beliebige ganze Zahl • `30000` (Standard): Standardwert von 30 Sekunden
`responseDelay`	Antwort-Verzögerung in Millisekunden **Mögliche Werte:** • Beliebige ganze Zahl • `750` (Standard): Standardwert von 0.75 Sekunden
`cleanAnsiVTSequences`	Angabe, ob die Ausgabe ANSI-VT-Zeichen enthalten soll oder nicht **Mögliche Werte:** • `true` (Standard): Die Ausgabe enthält ANSI-VT-Zeichen • `false`: Die Ausgabe enthält keine ANSI-VT-Zeichen

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt
`-1`	Bei der Ausführung des Adapters kam es zu einem FehlerThe adapter operation was executed successfully

Input

Bei der Verbindungsherstellung über Telnet, verwendet der Adapter standardmäßig Labels innerhalb der Zugangsdaten `login:` und `password:`. Wenn Sie jedoch eine Verbindung zu einem Rechner

herstellen möchten, der abweichende Zugangs-Labels hat, können diese im Input-XML über den Knoten `<Configuration>` konfiguriert werden.

Der `Telnet Connector` erwartet folgende Input-Struktur:

Input

```
<Telnet> <!-- Any root name respecting XML naming rules -->
    <!-- Optional attributes. If ignored, the default patterns will be used. -->
    <Configuration loginLabel="Username: " passwordLabel="Password: "/>
    <Commands>
        <Command value="ls -ll" />
        <Command value="mkdir /home/luis/myNewFolder"/>
        <Command value="touch /home/luis/myNewFolder/aNewEmptyFile.txt"/>
        <Command value="ls -ll"/>
    </Commands>
    <!-- Any number of Command nodes, which will be executed in the same connection
-->
</Telnet>
```

Erläuterung:

- `Telnet`: Wurzelknoten
- `Configuration`: Knoten mit demselben Inhalt wie im Input angegeben. Obwohl der Knoten erforderlich ist, sind die Attribute `loginLabel` und `passwordLabel` optional.
- `Commands`: Knoten, der alle verwendeten Befehle umschließt
- `Command`: Beliebige Anzahl an `Command`-Knoten mit einem `value`-Attribut, der die auszuführenden Befehle enthält.

Ausgabe

Der `Telnet Connector` gibt eine XML-Struktur aus, die folgender Ausgabe ähnelt:

Output

```
<Telnet>
    <LoginResult>Successful</LoginResult>
    <Configuration loginLabel="Username: " passwordLabel="Password: "/>
    <Commands>
        <Command value="ls -ll">total 52
            -rw-rw-r-- 1 ivan ivan 555 feb 24 15:30 a.txt
            drwxr-xr-x 2 ivan ivan 4096 feb 20 12:59 Desktop
            drwxr-xr-x 2 ivan ivan 4096 feb 20 12:59 Documents
            drwxr-xr-x 2 ivan ivan 4096 feb 20 12:59 Downloads
            -rw-r--r-- 1 ivan ivan 8980 feb 20 12:45 examples.desktop
            drwxrwxr-x 2 ivan ivan 4096 feb 21 14:54 mio
            drwxr-xr-x 2 ivan ivan 4096 feb 20 12:59 Music
            drwxr-xr-x 2 ivan ivan 4096 feb 20 12:59 Pictures
            drwxr-xr-x 2 ivan ivan 4096 feb 20 12:59 Public
            drwxr-xr-x 2 ivan ivan 4096 feb 20 12:59 Templates
            drwxr-xr-x 2 ivan ivan 4096 feb 20 12:59 Videos
            $
        </Command>
    </Commands>
    <!-- Command node containing the value (response) for each Command node available
in the input file -->
</Telnet>
```

Erläuterung:

- `Telnet`: Wurzelknoten
- `LoginResult`: Information zum Login-Ergebnis. Der Wert des Knotens ist entweder `Failed` od er `Successful`.
- `Configuration`: Knoten mit demselben Inhalt wie im Input angegeben.
- `Commands`: Knoten, der alle verwendeten Befehle umschließt
- `Command`: Beliebige Anzahl an `Command`-Knoten mit einem `value`-Attribut, der die auszuführenden Befehle enthält. Der Inhalt dieses Knotens ist das vom Telnet-Server ausgegebene Ergebnis des Befehls.

10.5 Compression

- gzip Converter
- ZIP

10.5.1 gzip Converter

? Unknown Attachment `gzip Converter`: Komprimiert und dekomprimiert Input-Daten mit dem *gzip*-Algorithmus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *Encode*: Daten gzip-komprimieren • *Decode*: gzip-komprimierte Daten als MIME-Typ *Application/Octet- stream* dekomprimieren

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.GZipAdapter*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Siehe auch:

- ZIP in Memory Compression (veraltet)
- Function Adapter

10.5.2 ZIP

10.5.2.1 Zip Converter

`Zip Converter`: Kann Dateien und Ordner im Speicher oder im Dateisystem zippen und entzippen.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Zip`: Komprimiert den bereitgestellten Input in einer Zip-Datei. Es können Verschlüsselungs- und Komprimierungseinstellungen vorgenommen werden. **Erforderliche Parameter** Die Parameter `SourcePath` oder `InputName` werden für diesen Vorgang benötigt. • `Unzip`: Dekomprimiert den bereitgestellten Input am angegebenen Speicherort. Es können auch passwortgeschützte Dateien dekomprimiert werden.

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.adapter.zipconverter.ZipConverter*: Hauptklasse (Standard)
`sourcePath`	Verzeichnispfad zur Datei, die komprimiert/dekomprimiert werden soll. **Mögliche Werte:** • Verzeichnispfad Wenn dieser Parameter nicht gesetzt wird, dann wird die Datei über den Adapter-Input bereitgestellt. In diesem Fall ist der Parameter `inputName` für die Operation `Zip` erforderlich.
`targetPath`	Verzeichnispfad zum Speicherort, an dem der Output gespeichert werden soll ("Zielverzeichnis"). **Mögliche Werte:** • Verzeichnispfad Wenn dieser Parameter nicht gesetzt wird, dann wird die Datei über den Adapter-Output bereitgestellt.

`overwrite`	Legt fest, ob vorhandene Dateien mit dem gleichen Namen überschrieben werden sollen. **Mögliche Werte:** • Aktiviert ⓘ • Wenn `mergeContent` aktiviert ist, dann wird im Output eine Fehlermeldung ausgegeben, da die Datei nicht weiterverarbeitet werden kann. • Wenn `mergeContent` deaktiviert ist, dann wird die vorhandene Datei überschrieben. • Deaktiviert ⓘ • Wenn `mergeContent` aktiviert ist, dann wird werden die Dateien zusammengeführt. • Wenn `mergeContent` deaktiviert ist, dann wird im Output eine Fehlermeldung ausgegeben, da die Datei nicht weiterverarbeitet werden kann.
`mergeContent`	Legt fest, ob vorhandene Dateien mit dem gleichen Namen zusammengeführt werden sollen. **Mögliche Werte:** • Aktiviert ⓘ • Wenn `overwrite` aktiviert ist, dann wird im Output eine Fehlermeldung ausgegeben, da die Datei nicht weiterverarbeitet werden kann. • Wenn `overwrite` deaktiviert ist, dann werden die Dateien zusammengeführt. • Deaktiviert ⓘ • Wenn `overwrite` aktiviert ist, dann wird die vorhandene Datei überschrieben. • Wenn `overwrite` deaktiviert ist, dann wird im Output eine Fehlermeldung ausgegeben, da die Datei nicht weiterverarbeitet werden kann.
`inputName`	Name der Input-Datei. Wird verwendet, wenn der `sourcePath` nicht gesetzt ist. **Mögliche Werte:** • Dateiname der Datei, die als Input empfangen bzw. als Output ausgegeben werden soll

`encryptedFiles`	Gibt an, ob in der Operation verschlüsselte Dateien verarbeitet werden sollen. Bei der Operation `Zip` wird die ausgegebene Zip-Datei passwortgeschützt, bei der Operation `Unzip` wird für das entzippen ein Passwort benötigt. **Mögliche Werte:** • Aktiviert / Deaktiviert
`password`	Passwort für verschlüsselte Dateien. **Mögliche Werte:** Beliebige Zeichenkette
`encryptionMethod`	Gibt an, welche Verschlüsselungsmethode beim Zippen verwendet wird, wenn `encryptedFiles` aktiviert ist. **Mögliche Werte:** • `No encryption`: Keine Verschlüsselung beim Zippen verwenden • `AES`: AES-Verschlüsselungsalgorithmus verwenden (Schlüssellänge: 256 Bit) • `Standard`: Standard-Verschlüsselungsalgorithmus verwenden
`compressionMethod`	Gibt an, welcher Komprimierungsalgorithmus beim Zippen verwendet wird. **Mögliche Werte:** • `Deflate` (Standard): Verlustfreie Komprimierung, die eine Kombination aus dem LZ77-Algorithmus und Huffman-Codierung verwendet. Weitere Informationen zu dieser Kompressionsstufe hier und hier. ⓘ Wenn `Deflate` ausgewählt ist, dann kann das Komprimierungslevel mit `CompressionLevel` spezifiziert werden. • `Store`: Daten werden nicht komprimiert, sondern in eine einzige Datei gepackt. ⓘ Das ist nützlich wenn mehrere Daten/Ordner gesendet/empfangen werden sollen und die Dateigröße vernachlässigt werden kann. Es wird keine Zeit für das Verpacken benötigt.
`compressionLevel`	Gibt den Komprimierungsalgorithmus beim Zippen an, wenn für `ompressionMethod` `Deflate` gewählt ist. **Mögliche Werte:** • `Fastest`: Niedrigste Kompressionsstufe, aber sehr schnelle Kompression • `Fast`: Niedrige Kompressionsstufe, aber schnelle Kompression • `Normal` (Standard): Optimales Verhältnis zwischen Kompressionsstufe und -geschwindigkeit • `Maximum`: Hohe Kompressionsstufe, aber langsamere Kompression • `Ultra`: Höchste Kompressionsstufe, aber sehr langsame Kompression

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters ist ein Fehler aufgetreten.

Eingabe

Der Adapter erwartet Input in folgender Struktur:

- Beide Operationen (`Zip` / `Unzip`) können mit zwei verschiedenen Herkunfts- bzw. Zielorten arbeiten:
  - Input/Output
  - Verzeichnispfad für Input/Output
- Basierend auf diesen zwei Möglichkeiten können für beide Operationen verschiedene Inputs verwendet werden:
  - Parameter `sourcePath` ist gesetzt: Input vom Verzeichnis, dass im Verzeichnispfad angegeben ist.
  - Parameter `sourcePath` ist nicht gesetzt: Input des Parameters
- Basierend auf diesen zwei Möglichkeiten können für beide Operationen verschiedene Outputs verwendet werden:
  - Parameter `targetPath` ist gesetzt: Das Ergebnis der Operation wird am Speicherort abgelegt, der als Verzeichnispfad angegeben ist.
  - Parameter `targetPath` ist nicht gesetzt:
    - Eine XML-Datei mit einem Eintrag für jede Input-Datei, die in der Input-Zip-Datei enthalten ist, wird ausgegeben.
    - Der Inhalt der Zip-Datei wird am Speicherort abgelegt, der als Verzeichnispfad angegeben ist.

Output

Der Adapter liefert abhängig von der ausgeführten Operation unterschiedliche Arten von Dokumenten zurück.

- `Zip`-Operation ohne `targetPath`: Zip-Datei wird im Adapter-Output bereitgestellt.
- `Zip`-Operation mit `targetPath`: Zip-Datei wird im Zielverzeichnis abgelegt.
- `Unzip`-Operation ohne `targetPath`: Eine XML-Datei, die die verschiedenen Dateien innerhalb der Zip-Datei enthält, wird im Adapter-Output bereitgestellt. Die XML-Datei hat folgende Struktur:

  Beispiel-Output für die Operation Unzip

  ```
  <?xml version="1.0" encoding="UTF-8" ?>
  <Zip>
      <File size="209" name="testToZip.xml" comment="&#0;&#0;!
  >PD94bWwgdmVyc2lvgj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPFJvb3RFbGVtZW50P</File>
  </Zip>
  ```

- `Unzip`-Operation mit `targetPath`: Der Inhalt der Zip-Datei wird im Zielverzeichnis abgelegt.

10.6 Datenbanken

- JDBC Connector
- JDBC Iterator
- JDBC Metadata Explorer

10.6.1 JDBC Connector

`JDBC Connector`: Stellt eine Verbindung zu einer JDBC-kompatiblen Datenbank her und führt ein oder mehrere Datenbank-Statements aus, die als Input-XML-Dokument mit adapterspezifischen Strukturen übergeben wurden.

Ehemalige Bezeichnung bis Version 4.5.1: *JDBC Database Connection Adapter*

> Voraussetzung ist eine im Applikations-Server bereitgestellte JDBC-DataSource, für welche die Verbindungs-URL, Zugangsdaten, Schema sowie Transaktionsverhaltensweisen etc. definiert wurden. Nähere Informationen zum Bereitstellen von JDBC-DataSources entnehmen Sie der Dokumentation Ihres Applikations-Servers.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** `execute`: Funktions-Adapter ausführen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.jdbc.DB*: Hauptklasse (Standard)
JndiName	JNDI-Name der DataSource Wenn die Verbindung über den JNDI-Namen hergestellt wird, dann darf der Parameter `connectionURL` nicht ausgefüllt werden. **Mögliche Werte:** Gültiger JNDI-Name (z. B. *java:/DefaultDS*)

ColumnCase	Steuert wie die Spaltennamen ins Ergebnis übernommen werden. Da Datenbanken unterschiedliches Verhalten betreffend der Groß-/Kleinschreibung von Spaltennamen haben, kann man hiermit die Weiterverarbeitung des Ergebnis unabhängig vom Datenbankverhalten machen. **Mögliche Werte:** • *NO_CHANGE*: Ändert die Spaltennamen nicht, d.h. die Schreibweise im Ergebnis ist exakt wie von der Datenbank geliefert (Standard) • *TO_LOWER*: Ändert die Spaltennamen auf Kleinbuchstaben • *TO_UPPER*: Ändert die Spaltennamen auf Großbuchstaben ⓘ Spaltenname, die keine gültigen Elementnamen (`NCNAME`) darstellen, werden durch `_<number>` (e. g. `_2`) ersetzt, wo *<number>* der Index der Spalte ist.
connectionURL	Verbindungs-URL für die JDBC-Datenbank ℹ Wenn die Verbindung über die Verbindungs-URL hergestellt wird, dann darf der Parameter `JndiName` nicht ausgefüllt werden. **Mögliche Werte:** • Die Art des Aufbaus hängt von der Datenbank ab, zu der eine Verbindung aufgebaut werden soll. • Beispiel für sqlite: *`jdbc:sqlite:C:/Temp/mydatabase.db`*
user	Benutzername zur Authentifizierung bei Verwendung des Parameters `connectionURL`
password	Passwort zur Authentifizierung bei Verwendung des Parameters `connectionURL`
rootElementName	Name des Wurzelelements des Outputs. **Mögliche Werte:** Zeichenkette, die für XML-Elementnamen gültig ist. Standard: *`Data`*

CommitMode	Steuert, welche Variante für Commits verwendet werden soll **Mögliche Werte:** • *NONE*: Keinen Commit durchführen • *AUTO*: Automatischer Commit gemäß JDBC-API (d. h. nach jedem Batch) (Standard) • *STATEMENT*: Explizites Commit nach jedem Statement (d. h. nach allen Batches für dieses Statement) • *DOCUMENT*: Am Ende des Dokuments (d. h. nach Ausführung aller Statements) • *MANAGED*: Adapter in Transaktionen nutzen (für extern verwaltete Transaktionen)
AddMetaDataForQueries	(ab *X4* 4.1 Revision *32408*) Steuert, ob Datenbank-Metadaten, siehe JDBC Metadata Explorer, zu jedem *Result Set* im Ergebnis-XML-Dokument des Adapters ausgegeben werden sollen, wenn der Adapter eine SQL-Abfrage ausführt (nicht der Fall bei `UPDATE`, `DELETE` etc.). Im Batch-Modus (`StatementBatch`, `PreparedStatementBatch` und `CallableStatementBatch`) ist dies ebenfalls nicht möglich. **Mögliche Werte:** • *ja*: Dem Ergebnisdokument Datenbank-Metadaten hinzufügen • *nein*: Ergebnisdokument ohne Datenbank-Metadaten ausgeben (Standard)

Statuswerte

1	Die Abfrage hat Datensätze geliefert bzw. die Anweisung hat Datensätze geändert Der Status 1 wird geliefert, wenn mehrere Abfragen bzw. Anweisungen im Dokument enthalten sind und mindestens eine Abfrage ein Ergebnis geliefert oder verändert hat.
0	Die Abfrage hat keine Ergebnisse geliefert oder die Anweisung hat keine Datensätze geändert Der Status `0` wird geliefert, wenn mehrere Abfragen bzw. Anweisungen im Dokument enthalten sind und keine Abfrage ein Ergebnis geliefert oder verändert hat.
-1	Technischer Fehler: genauere Informationen zur Fehlerursache entnehmen Sie dem Server-Log

Input

Der *JDBC Connector* erwartet spezifische Input-XML-Strukturen, die Datenbank-Statements enthalten. Diese Input-XML-Strukturen können über XSL-Mappings in Abhängigkeit von den verarbeiteten XML-Daten dynamisch generiert werden.

Siehe auch:

- Statement: Direktes SQL
- StatementBatch: Direktes SQL im Batch-Modus
- PreparedStatement: SQL mit Werten
- PreparedStatementBatch: SQL mit Werten im Batch-Modus
- CallableStatement: Stored Procedures
- CallableStatementBatch: Stored Procedures im Batch-Modus
- NamedPreparedStatement: SQL mit benannten Parametern
- NamedPreparedStatementBatch: SQL mit benannten Parametern im Batch-Modus
- Durchleiten von XML-Strukturen
- Abrufen von Datenbank-Metadaten

10.6.1.1 Statement: Direktes SQL

Input

```
<Beliebiger Elementname>
   <statement resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
         SQL-Anweisungen
      </sql>
   </statement>
</Beliebiger Elementname>
```

Für jedes Element `<statement/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2* .
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.
- Es enthält mindestens ein Element `<sql/>`, das jeweils eine SQL-Anweisung enthält.

> Falls Sie innerhalb der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA* -Block, z. B. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]>`

Beispiel

```
<select>
   <statement resulttag="result" timezone="UTC" format="dd.MM.yyyy">
      <sql>
         <![CDATA[SELECT * FROM JMS_USERS]]>
      </sql>
   </statement>
</select>
```

Output

Wurde die SQL-Anweisung erfolgreich ausgeführt, gibt der Datenbankverbindungs-Baustein ein XML-Dokument mit folgender Struktur zurück, das die Antwort der Datenbank als Zeichenkette (String) enthält:

```
<Data>
   <Ergebniselementname>
      <Feld1><!--Wert--></Feld1>
      <Feldn><!--Wert--></Feldn>
   </Ergebniselementname>
</Data>
```

> Bei `SELECT`-Statements zum Zugriff auf Datenbanktabellen mit Spaltennamen, die keine gültigen XML-Namen darstellen, werden im Ergebnis-XML-Dokument synthetische Elementnamen erzeugt, um die Wohlgeformheit des XML-Dokuments sicherzustellen.

10.6.1.2 StatementBatch: Direktes SQL im Batch-Modus

Input

Der *JDBC Connector* führt mehrere SQL-Anweisungen auf einmal im Batch-Modus aus, wenn sie in folgender Input-XML-Struktur abgelegt werden. Dabei sind lediglich SQL-Anweisungen möglich, die kein ResultSet zurückliefern, also z. B. keine Abfragen mit `SELECT`:

```
<Beliebiger Elementname>
   <statementbatch resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
         SQL-Anweisung 1
      </sql>
      <sql>
         SQL-Anweisung n
      </sql>
   </statementbatch>
</Beliebiger Elementname>
```

Für jedes Element `<statementbatch/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.
- Es enthält mindestens ein Element `<sql/>`, das jeweils eine SQL-Anweisung enthält.

> Falls Sie innerhalb der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=100 ]]>`

Beispiel

```
<select>
   <statementbatch resulttag="Ergebniselementname" timezone="UTC" format="dd.MM.yyyy">
      <sql>
         <![UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=100]]>
      </sql>
      <sql>
         <![UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=200]]>
      </sql>
   </statementbatch>
</select>
```

Output

Wurde die Batch-Anweisung erfolgreich ausgeführt, gibt der Datenbankverbindungs-Baustein folgende XML-Struktur zurück:

```
<Data>
   <Ergebniselementname>
      <!--Status der Batch-Operation-->
   </Ergebniselementname>
</Data>
```

Als Status der Batch-Operation sind folgende Ausgaben möglich:

- *(Anzahl der Zeilen)* `rows affected.` oder
- `SUCCESS_NO_INFO` oder
- `EXECUTE_FAILED`

10.6.1.3 PreparedStatement: SQL mit Werten

Der *JDBC Connector* führt SQL-Anweisungen mit dynamischen Input-Daten aus, wenn sie in folgender Input-XML-Struktur abgelegt werden.

> Alternativ können Sie ab Release 5.0 der X4 BPMS auch SQL-Anweisungen mit benannten Parametern ausführen, siehe NamedPreparedStatement: SQL mit benannten Parametern.

Input

```
<Beliebiger Elementname>
   <preparedstatement resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
         <!--SQL-Anweisungen-->
      </sql>
      <columns>
         <column mode="in" name="frei wählbarer NCName"
          index="Platzhalter-Index" type="Datentyp"
          timezone="Zeitzone" format="Datumsformat">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--Wert--></Name>
         </row>
      </rowset>
   </preparedstatement>
</Beliebiger Elementname>
```

Für jedes Element `<preparedstatement/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß *java.text.SimpleDateFormat* an.
- Es enthält mindestens ein Element `<sql/>`.

> Falls Sie innerhalb der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]>`

Für jedes Element `<column/>` gilt:

- Es darf beliebig häufig verwendet werden (*0 – n*-mal)
- Das optionale Attribut `mode` gibt die "Richtung" des Datenstroms an (*in*: Input-Parameter für die Datenbank; *out*: Ergebnisparameter aus der Datenbank; *inout*: sowohl Input-Parameter als auch Ergebnisparameter; Standard: *in*.
- Das Attribut `name` gibt einen frei wählbaren Namen (*NCName*) für ein Parameter an.
- Das Attribut `index` gibt einen ganzzahligen Platzhalterindex für den in Attribut `name` definierten Parameter an.
- Das optionale Attribut `type` gibt den Datentyp an (z. B. *INTEGER* oder *BLOB*; Standard: *VARCHAR*) .
- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.

Für jedes Element `<rowset/>` gilt:

- Es darf beliebig häufig verwendet werden (*0 – n*-mal)
- Es enthält mindestens ein Element `<row/>`, das die Statement-Ausführung steuert.

Für jedes Element `<row/>` gilt:

- Es darf beliebig viele Elemente enthalten, die den Namen der Tabellenspalte oder den im SQL-Statement gesetzten Spalten-Alias tragen.
- Die darin liegenden Elemente enthalten die jeweiligen Werte.

Beachten Sie Folgendes

- Die Attribute `mode` und `format` sind in *X4 Server* Revision 31469 Pflichtparameter; ab neueren Versionen sind diese optional.
- Zulässige Datentypen für das Attribut `type` zum Setzen (Input-Parameter) und Lesen (*ResultSet*) von Werten sind: *BIGINT, BINARY, BIT, BOOLEAN, BLOB, CHAR, CLOB, DATE, DECIMAL, DOUBLE, FLOAT, INTEGER, LONGNVARCHAR, LONGVARBINARY, LONGVARCHAR, NCHAR, NCLOB, NULL, NUMERIC, NVARCHAR, REAL, SMALLINT, SQLXML, TIME, TIMESTAMP, TINYINT, VARBINARY und VARCHAR* (Standard), *XML* (alias für SQLXML).

Beispiel

```
<Rootelement>
   <preparedstatement resulttag="abc" timezone="MEZ"
      format="dd.MM.yyyy">
      <sql>
      <![CDATA[SELECT * FROM JMS_USERS WHERE USERID=?]]>
      </sql>
      <columns>
         <column mode="in" name="id" index="1"
            type="INTEGER">
         </column>
      </columns>
      <rowset>
         <row>
            <id>23</id>
         </row>
      </rowset>
   </preparedstatement>
</Rootelement>
```

Output

Wurde die SQL-Anweisung erfolgreich ausgeführt, gibt der Adapter ein XML-Dokument mit folgender Struktur zurück, das die Antwort der Datenbank als Zeichenkette (String) enthält:

```
<Data>
   <Ergebniselementname>
      <Feld1><!--Wert--></Feld1>
      <Feldn><!--Wert--></Feldn>
   </Ergebniselementname>
</Data>
```

10.6.1.4 PreparedStatementBatch: SQL mit Werten im Batch-Modus

Der *JDBC Connector* führt mehrere SQL-Anweisungen auf einmal im Batch-Modus aus, wenn sie in folgender Input-XML-Struktur abgelegt werden. Dabei sind lediglich SQL-Anweisungen möglich, die kein Result Set zurückliefern, also z.B. keine Abfragen mit `SELECT`.

> ⓘ Alternativ können Sie ab Release 5.0 der X4 BPMS auch SQL-Anweisungen mit benannten Parametern mit oder ohne Batch-Modus ausführen, siehe NamedPreparedStatementBatch: SQL mit benannten Parametern im Batch-Modus.

Input

```
<Beliebiger Elementname>
   <preparedstatementbatch resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
         <!-- SQL-Anweisung -->
      </sql>
      <columns>
         <column mode="in" name="frei wählbarer NCName"
          index="Platzhalter-Index" type="Datentyp"
          timezone="Zeitzone" format="Datumsformat">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--Wert--></Name>
         </row>
         <!-- Weitere <row>-Elemente für Batch-Modus -->
      </rowset>
   </preparedstatementbatch>
</Beliebiger Elementname>
```

Für jedes Element `<preparedstatementbatch/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *`GMT+2`*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß *java.text.SimpleDateFormat* an.
- Es enthält mindestens ein Element `<sql/>`.

> ⓘ Falls Sie innerhalb der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=? ]]>`

Für jedes Element `<column/>` gilt:

- Es darf beliebig häufig verwendet werden (*0* – *n*-mal)
- Das optionale Attribut `mode` gibt die "Richtung" des Datenstroms an (*`in`*: Input-Parameter für die Datenbank; *`out`*: Ergebnisparameter aus der Datenbank; *`inout`*: sowohl Input-Parameter als auch Ergebnisparameter; Standard: *`in`*.
- Das Attribut `name` gibt einen frei wählbaren Namen (*NCName*) für ein Parameter an.
- Das Attribut `index` gibt einen ganzzahligen Platzhalterindex für den in Attribut `name` definierten Parameter an.
- Das optionale Attribut `type` gibt den Datentyp an (z. B. *`INTEGER`* oder *`BLOB`*; Standard: *`VARCHAR`*) .
- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *`GMT+2`*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.

Für jedes Element `<rowset/>` gilt:

- Es darf beliebig häufig verwendet werden (*0 – n*–mal)
- Es enthält beliebig viele Elemente `<row/>`, welche die Statement-Ausführung steuern.

Für jedes Element `<row/>` gilt:

- Es darf beliebig viele Elemente enthalten, die den Namen der Tabellenspalte oder den im SQL-Statement gesetzten Spalten-Alias tragen.
- Die darin liegenden Elemente enthalten die jeweiligen Werte.

Beachten Sie Folgendes

- Die Attribute `mode` und `format` sind in *X4 Server* Revision 31469 Pflichtparameter; ab neueren Versionen sind diese optional.
- Zulässige Datentypen für das Attribut `type` zum Setzen (Input-Parameter) und Lesen (*ResultSet*) von Werten sind: *BIGINT, BINARY, BIT, BOOLEAN, BLOB, CHAR, CLOB, DATE, DECIMAL, DOUBLE, FLOAT, INTEGER, LONGNVARCHAR, LONGVARBINARY, LONGVARCHAR, NCHAR, NCLOB, NULL, NUMERIC, NVARCHAR, REAL, SMALLINT, SQLXML, TIME, TIMESTAMP, TINYINT, VARBINARY* und *VARCHAR* (Standard), *XML* (alias für SQLXML).

Beispiel

```
<Rootelement>
   <preparedstatementbatch resulttag="Ergebniselementname" timezone="MEZ"
      format="dd.MM.yyyy">
      <sql>
      <![CDATA[ UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=? ]]>
      </sql>
      <columns>
         <column mode="in" name="id" index="1"
            type="INTEGER">
         </column>
      </columns>
      <rowset>
         <row>
            <id>23</id>
         </row>
         <row>
            <id>25</id>
         </row>
      </rowset>
   </preparedstatementbatch>
</Rootelement>
```

Output

Wurde die Batch-Anweisung erfolgreich ausgeführt, gibt der Datenbankverbindungs-Baustein folgende XML-Struktur zurück:

```
<Data>
   <Ergebniselementname>
      <!--Status der Batch-Operation-->
   </Ergebniselementname>
</Data>
```

Als Status der Batch-Operation sind folgende Ausgaben möglich:

- *(Anzahl der Zeilen)* `rows affected.` oder
- `SUCCESS_NO_INFO` oder
- `EXECUTE_FAILED`

10.6.1.5 CallableStatement: Stored Procedures

Der *JDBC Connector* führt Stored Procedures über *JDBC Callablestatements* mit dynamischen Input-Daten aus, wenn sie in folgender Input-XML-Struktur abgelegt werden.

Input

```
<Beliebiger Elementname>
   <callablestatement resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
        <!--SQL-Anweisungen -->
      </sql>
      <columns>
         <column mode="Modus" name="frei wählbarer Name" index="Platzhalter"
          type="Datentyp" timezone="Zeitzone" format="Datumsformat">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--Wert--></Name>
         </row>
      </rowset>
   </callablestatement>
</Beliebiger Elementname>
```

Für jedes Element `<callablestatement/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.
- Es enthält mindestens ein Element `<sql/>`.

Falls Sie innerhalb der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]>`

Für jedes Element `<column/>` gilt:

- Es darf beliebig häufig verwendet werden (*0* – *n*-mal)
- Das optionale Attribut `mode` gibt die "Richtung" des Datenstroms an (*in*: Input-Parameter für die Datenbank; *out*: Ergebnisparameter aus der Datenbank; *inout*: sowohl Input-Parameter als auch Ergebnisparameter; siehe http://download.oracle.com/javase/1,5.0/docs/guide/jdbc/getstart/callablestatement.html); Standard: *in*.
- Das Attribut `name` gibt einen frei wählbaren Namen für ein Parameter an.
- Das Attribut `index` gibt einen ganzzahligen Platzhalterindex für den in Attribut `name` definierten Parameter an.
- Das optionale Attribut `type` gibt den Datentyp an (zulässige Datentypen siehe Hinweis unten; Standard: *VARCHAR*).
- Das Attribut `name` gibt einen frei wählbaren Namen für ein Parameter an.
- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.

Für jedes Element `<rowset/>` gilt:

- Es darf beliebig häufig verwendet werden (*0* – *n*-mal).
- Es enthält beliebig viele Elemente `<row/>`, die die Statement-Ausführung steuern.

Für jedes Element `<row/>` gilt:

- Es darf beliebig viele Elemente enthalten, die den Namen der Tabellenspalte oder den im SQL-Statement gesetzten Spalten-Alias tragen.
- Die darin liegenden Elemente enthalten die jeweiligen Werte.

Beachten Sie Folgendes

- Die Attribute `mode` und `format` sind in *X4 Server* Revision 31469 Pflichtparameter; ab neueren Versionen sind diese optional.
- Zulässige Datentypen für das Attribut `type` zum Setzen (Input-Parameter) und Lesen (*ResultSet*) von Werten sind: *BIGINT, BINARY, BIT, BOOLEAN, BLOB, CHAR, CLOB, DATE, DECIMAL, DOUBLE, FLOAT, INTEGER, LONGNVARCHAR, LONGVARBINARY, LONGVARCHAR, NCHAR, NCLOB, NULL, NUMERIC, NVARCHAR, REAL, SMALLINT, SQLXML, TIME, TIMESTAMP, TINYINT, VARBINARY und VARCHAR* (Standard), *XML* (alias für SQLXML).

Beispiel

```
<Rootelement>
   <callablestatement resulttag="abc" timezone="MEZ"
      format="dd.MM.yyyy">
      <sql>
      <![CDATA[ {call getData(?)} ]]>
      </sql>
      <columns>
         <column mode="out" name="id" index="1"
            type="INTEGER">
         </column>
      </columns>
      <rowset>
         <row>
            <id>TestUser</id>
         </row>
      </rowset>
   </callablestatement>
</Rootelement>
```

Output

Wurde die SQL-Anweisung erfolgreich ausgeführt, gibt der Datenbankverbindungs-Baustein ein XML-Dokument mit folgender Struktur zurück, das die Antwort der Datenbank als Zeichenkette (String) enthält:

```
<Data>
   <Ergebniselementname>
      <Feld1><!--Wert--></Feld1>
      <Feldn><!--Wert--></Feldn>
   </Ergebniselementname>
</Data>
```

10.6.1.6 CallableStatementBatch: Stored Procedures im Batch-Modus

Der *JDBC Connector* führt Stored Procedures über *JDBC Callablestatements* mit dynamischen Input-Daten aus, wenn sie in folgender Input-XML-Struktur abgelegt werden. Dabei sind lediglich SQL-Anweisungen möglich, die kein ResultSet zurückliefern, also z. B. keine Abfragen mit `SELECT`.

Input

```
<Beliebiger Elementname>
   <callablestatementbatch resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
        <!--SQL-Anweisungen -->
      </sql>
      <columns>
         <column mode="Modus" name="frei wählbarer Name" index="Platzhalter"
          type="Datentyp" timezone="Zeitzone" format="Datumsformat">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--Wert--></Name>
         </row>
      </rowset>
   </callablestatementbatch>
</Beliebiger Elementname>
```

Für jedes Element `<callablestatementbatch/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.
- Es enthält mindestens ein Element `<sql/>`.

> Falls Sie innerhalb der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]`

Für jedes Element `<column/>` gilt:

- Es darf beliebig häufig verwendet werden (*0* – *n*-mal)
- Das optionale Attribut `mode` gibt die "Richtung" des Datenstroms an (*in*: Input-Parameter für die Datenbank; *out*: Ergebnisparameter aus der Datenbank; *inout*: sowohl Input-Parameter als auch Ergebnisparameter; siehe http://download.oracle.com/javase/1,5.0/docs/guide/jdbc/getstart/callablestatement.html); Standard: *in*.
- Das Attribut `name` gibt einen frei wählbaren Namen für ein Parameter an.
- Das Attribut `index` gibt einen ganzzahligen Platzhalterindex für den in Attribut `name` definierten Parameter an.
- Das optionale Attribut `type` gibt den Datentyp an (zulässige Datentypen siehe Hinweis unten; Standard: *VARCHAR*).
- Das Attribut `name` gibt einen frei wählbaren Namen für ein Parameter an.
- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.

- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.

Für jedes Element `<rowset/>` gilt:

- Es darf beliebig häufig verwendet werden (*0* – *n*-mal).
- Es enthält beliebig viele Elemente `<row/>`, die die Statement-Ausführung steuern.

Für jedes Element `<row/>` gilt:

- Es darf beliebig viele Elemente enthalten, die den Namen der Tabellenspalte oder den im SQL-Statement gesetzten Spalten-Alias tragen.
- Die darin liegenden Elemente enthalten die jeweiligen Werte.

Beachten Sie Folgendes

- Die Attribute `mode` und `format` sind in *X4 Server* Revision 31469 Pflichtparameter; ab neueren Versionen sind diese optional.
- Zulässige Datentypen für das Attribut `type` zum Setzen (Input-Parameter) und Lesen (*ResultSet*) von Werten sind: *`BIGINT`, `BINARY`, `BIT`, `BOOLEAN`, `BLOB`, `CHAR`, `CLOB`, `DATE`, `DECIMAL`, `DOUBLE`, `FLOAT`, `INTEGER`, `LONGNVARCHAR`, `LONGVARBINARY`, `LONGVARCHAR`, `NCHAR`, `NCLOB`, `NULL`, `NUMERIC`, `NVARCHAR`, `REAL`, `SMALLINT`, `SQLXML`, `TIME`, `TIMESTAMP`, `TINYINT`, `VARBINARY` und `VARCHAR`* (Standard), *`XML`* (alias für SQLXML).

Beispiel

```
<Rootelement>
   <callablestatementbatch resulttag="abc" timezone="MEZ"
      format="dd.MM.yyyy">
      <sql>
      <![CDATA[ {call getData(?)} ]]>
      </sql>
      <columns>
         <column mode="out" name="id" index="1"
            type="INTEGER">
         </column>
      </columns>
      <rowset>
         <row>
            <id>TestUser</id>
         </row>
      </rowset>
   </callablestatementbatch>
   <callablestatementbatch resulttag="abc" timezone="MEZ"
      format="dd.MM.yyyy">
      <sql>
      <![CDATA[ {call getData(?)} ]]>
      </sql>
      <columns>
         <column mode="out" name="id" index="1"
            type="INTEGER">
         </column>
      </columns>
      <rowset>
         <row>
            <id>TestUser</id>
         </row>
      </rowset>
   </callablestatementbatch>
</Rootelement>
```

Output

Wurde die Batch-Anweisung erfolgreich ausgeführt, gibt der Datenbankverbindungs-Baustein folgende XML-Struktur zurück:

```
<Data>
   <Ergebniselementname>
      <!--Status der Batch-Operation-->
   </Ergebniselementname>
</Data>
```

Als Status der Batch-Operation sind folgende Ausgaben möglich:

- *(Anzahl der Zeilen)* `rows affected.` oder
- `SUCCESS_NO_INFO` oder
- `EXECUTE_FAILED`

10.6.1.7 NamedPreparedStatement: SQL mit benannten Parametern

Der *JDBC Connector* führt SQL-Anweisungen mit benannten Parametern aus, wenn sie in folgender Input-XML-Struktur abgelegt werden.

Input

```
<Beliebiger Elementname>
   <namedpreparedstatement resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
         <!--SQL-Anweisungen (optional innerhalb eines CDATA-Blocks) -->
         <!--Je Parameter ein XML-Element, gleichnamig zum entsprechenden Kindelement
in <row> -->
      </sql>
      <rowset>
         <row>
            <Name><!--Wert--></Name>
         </row>
      </rowset>
   </namedpreparedstatement>
</Beliebiger Elementname>
```

Für jedes Element `<namedpreparedstatement/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß *java.text.SimpleDateFormat* an und bestimmt das Format der zurückgegebenen Datumswerten.
- Es enthält ein Element `<sql/>`.

Für jedes XML-Element eines benannten Parameters innerhalb der SQL-Anweisung in Element `<sql/>` gilt:

- Der Elementname gibt den Parameter-Namen an, der mit entsprechenden Element innerhalb von `<row/>` übereinstimmt.
- Das optionale Attribut `mode` gibt die "Richtung" des Datenstroms an (*in*: Input-Parameter für die Datenbank; *out*: Ergebnisparameter aus der Datenbank; *inout*: sowohl Input-Parameter als auch Ergebnisparameter; Standard: *in*.
- Das optionale Attribut `type` gibt den Datentyp an, z. B. *INTEGER*, *BLOB* oder *VARCHAR*(Standard).
- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an und gibt das erwartete Format für den Wert innerhalb des entsprechenden `<row>`-Elements vor.

Falls Sie in der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ SELECT * FROM TABLE WHERE ID<]]><id type="INTEGER" />`.

Für jedes Element `<rowset/>` gilt:

- Es darf beliebig häufig verwendet werden (*0 – n*–mal)
- Es enthält beliebig viele Elemente `<row/>`, die die Statement-Ausführung steuern.

Für jedes Element `<row/>` gilt:

- Es muss mindestens ein Element enthalten, das dem Elementnamen im SQL-Statement entspricht.
- Die darin liegenden Elemente enthalten die jeweiligen Werte.

Beachten Sie für jedes Datenelement innerhalb von `<row/>`, dass im Elementinhalt keine unbeabsichtigten Zeilenumbrüche definiert wurden (z.B. durch Editor-Formatierung). Andernfalls würden diese Zeilenumbrüche in der erzeugten SQL-Anweisung als `\n` angefügt.

Beachten Sie Folgendes
Zulässige Datentypen für das Attribut `type` zum Setzen (Input-Parameter) und Lesen (*ResultSet*) von Werten sind: *BIGINT, BINARY, BIT, BOOLEAN, BLOB, CHAR, CLOB, DATE, DECIMAL, DOUBLE, FLOAT, INTEGER, LONGNVARCHAR, LONGVARBINARY, LONGVARCHAR, NCHAR, NCLOB, NULL, NUMERIC, NVARCHAR, REAL, SMALLINT, SQLXML, TIME, TIMESTAMP, TINYINT, VARBINARY und VARCHAR* (Standard), *XML* (alias für SQLXML).

Beispiel

```
<Rootelement>
    <namedpreparedstatement resulttag="namedPreparedStatementWithCData"
     timezone="UTC" format="dd.MM.yyyy">
        <sql>
             SELECT * FROM PROCESS WHERE X4_DURATION= <duration type="INTEGER"
                mode="in" /> AND X4_PROCESSNAME= <processname
                type="VARCHAR" mode="in" />
        </sql>
        <rowset>
            <row>
                <duration>101</duration>
                <processname>Project/Process/Path/Process.wrf</processname>
            </row>
        </rowset>
    </namedpreparedstatement>
</Rootelement>
```

Output

Wurde die SQL-Anweisung erfolgreich ausgeführt, gibt der Adapter ein XML-Dokument mit folgender Struktur zurück, das die Antwort der Datenbank als Zeichenkette (String) enthält:

```
<Data>
   <Ergebniselementname>
      <Feld1><!--Wert--></Feld1>
      <Feldn><!--Wert--></Feldn>
   </Ergebniselementname>
</Data>
```

10.6.1.8 NamedPreparedStatementBatch: SQL mit benannten Parametern im Batch-Modus

Der *JDBC Connector* führt mehrere SQL-Anweisungen auf einmal im Batch-Modus aus, wenn sie in folgender Input-XML-Struktur jeweils in einem `<namedpreparedstatementbatch>`-Block abgelegt werden. Dabei sind lediglich SQL-Anweisungen möglich, die kein ResultSet zurückliefern, also z. B. keine Abfragen mit `SELECT`.

Input

```
<Beliebiger Elementname>
   <namedpreparedstatementbatch resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
         <!--SQL-Anweisungen (optional innerhalb eines CDATA-Blocks) -->
         <!--Je Parameter ein XML-Element, gleichnamig zum entsprechenden Kindelement
in <row> -->
      </sql>
      <rowset>
         <row>
            <Name><!--Wert--></Name>
         </row>
         <!-- Weitere <row>-Elemente für Batch-Modus -->
      </rowset>
   </namedpreparedstatementbatch>
</Beliebiger Elementname>
```

Für jedes Element `<namedpreparedstatementbatch/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *`GMT+2`*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß *java.text.SimpleDateFormat* an und bestimmt das Format der zurückgegebenen Datumswerten.
- Es enthält mindestens ein Element `<sql/>`.

Für jedes XML-Element eines benannten Parameters innerhalb der SQL-Anweisung im Element `<sql/>` gilt:

- Der Elementname gibt den Parameter-Namen an, der mit entsprechenden Element innerhalb von `<row/>` übereinstimmt.
- Das optionale Attribut `mode` gibt die "Richtung" des Datenstroms an (*`in`*: Input-Parameter für die Datenbank; *`out`*: Ergebnisparameter aus der Datenbank; *`inout`*: sowohl Input-Parameter als auch Ergebnisparameter; Standard: *`in`*.
- Das optionale Attribut `type` gibt den Datentyp an, z. B. *`INTEGER`*, *`BLOB`* oder *`VARCHAR`* (Standard).
- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *`GMT+2`*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an und gibt das erwartete Format für den Wert innerhalb des entsprechenden `<row>`-Elements vor.

> Falls Sie in der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ SELECT * FROM TABLE WHERE ID<]]><id type="INTEGER" />`.

Für jedes Element `<rowset/>` gilt:

- Es darf beliebig häufig verwendet werden (*`0`*–*`n`*–`mal`)
- Es enthält beliebig viele Elemente `<row/>`, die die Statement-Ausführung steuern.

Für jedes Element `<row/>` gilt:

- Es muss mindestens ein Element enthalten, das dem Elementnamen im SQL-Statement entspricht.
- Die darin liegenden Elemente enthalten die jeweiligen Werte.

Beachten Sie für jedes Datenelement innerhalb von `<row/>`, dass im Elementinhalt keine unbeabsichtigten Zeilenumbrüche definiert wurden (z.B. durch Editor-Formatierung). Andernfalls würden diese Zeilenumbrüche in der erzeugten SQL-Anweisung als `\n` angefügt.

> **Beachten Sie Folgendes**
> Zulässige Datentypen für das Attribut `type` zum Setzen (Input-Parameter) und Lesen (*ResultSet*) von Werten sind: *`BIGINT, BINARY, BIT, BOOLEAN, BLOB, CHAR, CLOB, DATE, DECIMAL, DOUBLE, FLOAT, INTEGER, LONGNVARCHAR, LONGVARBINARY, LONGVARCHAR, NCHAR, NCLOB, NULL, NUMERIC, NVARCHAR, REAL, SMALLINT, SQLXML, TIME, TIMESTAMP, TINYINT, VARBINARY und VARCHAR`* (Standard), *`XML`* (alias für SQLXML).

Beispiel

```
<Rootelement>
    <namedpreparedstatementbatch resulttag="namedPreparedStatementBatchWithCData"
     timezone="UTC" format="dd.MM.yyyy">
        <sql>
             <![CDATA[UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=]]>
             <duration type="INTEGER" mode="inout" />
        </sql>
        <rowset>
            <row>
                <duration>101</duration>
            </row>
            <row>
                <duration>9</duration>
            </row>
        </rowset>
    </namedpreparedstatementbatch>
</Rootelement>
```

Output

Wurde die SQL-Anweisung erfolgreich ausgeführt, gibt der Adapter ein XML-Dokument mit folgender Struktur zurück:

```
<Data>
   <Ergebniselementname count="Anzahl der betroffenen Datenbankeinträge">
      <!--Status der Batch-Operation-->
   </Ergebniselementname>
</Data>
```

Als Status der Batch-Operation sind folgende Ausgaben möglich:

- *(Anzahl der Zeilen)* `rows affected.` oder
- `SUCCESS_NO_INFO` oder
- `EXECUTE_FAILED`

10.6.1.9 Durchleiten von XML-Strukturen

Der *JDBC Connector* ermöglicht das Durchleiten von XML-Strukturen. In einem Input-XML-Dokument können Sie dabei zusätzlich zu den SQL-Statement-Strukturen eigene XML-Strukturen definieren, die der Adapter direkt wieder ausgibt. Damit lässt sich in vielen Fällen die Prozessmodellierung vereinfachen und die XML-Strukturen können gemeinsam mit den Ergebnissen der SQL-Anweisung(en) weiterverarbeitet werden.

Input

Im Input-XML-Dokument können Sie mit folgendem Statement, das als Kind des Wurzel-Elements definiert wird, eine oder mehrere XML-Struktur(en) durchleiten:

```
<copy resulttag="Name des Elternelements">
  Beliebige XML-Struktur
</copy>
```

Output

Die XML-Struktur, die innerhalb des `<copy>`-Elements definiert wurde, wird vom *JDBC Connector* ausgegeben, jedoch ohne Processing Instructions und Kommentare. Dabei wird die XML-Struktur von einem Element umschlossen, das in der Input-Struktur im Attribut `resulttag` angegeben wurde.

```
<Name des Elternelements>
  Input-XML-Struktur
</Name des Elternelements>
```

Beispiel

Im folgenden Input-XML-Dokument wird neben einem SQL-Statement eine XML-Struktur zur Durchleitung definiert:

```
<Root>
  <statement resulttag="result" timezone="UTC" format="dd.MM.yyyy">
    <sql>  <![CDATA[SELECT * FROM JMS_USERS]]>  </sql>
  </statement>
  <copy resulttag="Content">
    <Data>Text</Data>
  </copy>
</Root>
```

Der *JDBC Connector* gibt daraufhin z. B. folgendes Ergebnis-XML-Dokument aus, das die XML-Struktur mit dem angegebenen Elternelement enthält:

```
<Data>
  <result>
    <USERID>user</USERID>
    <PASSWD>password</PASSWD>
    <CLIENTID isNull="true" />
  </result>
  <Content>
    <Data>Text</Data>
  </Content>
</Data>
```

10.6.1.10 Abrufen von Datenbank-Metadaten

Mit dem Adapter-Parameter `AddMetaDataForQueries` (ab *X4 Server* Revision *32408*) können Sie steuern, ob Datenbank-Metadaten zu den entsprechenden *Result Sets* im Ergebnis-XML-Dokument

innerhalb eines XML-Elements ermittelt und gemeinsam mit den Ergebnissen der SQL-Abfrage ausgegeben werden sollen.

Der Name jedes XML-Elements, das die entsprechenden Metadaten enthält, setzt sich aus dem Wert des Attributs `resulttag` im Input-Dokument und dem Zusatz `-meta` zusammen, z. B. `<result-meta>`.

> Bei SQL-Statements, die kein *Result Set* ausgeben (z. B. bei `UPDATE` oder `DELETE`), können keine Metadaten ausgegeben werden. Im Batch-Modus des Adapters (`StatementBatch`, `PreparedStatementBatch` und `CallableStatementBatch`) ist dies ebenfalls nicht möglich.

Beispiel

Im folgenden Input-XML-Dokument wird eine SQL-Abfrage und als Ergebniselement der Name `result` definiert:

```
<Root>
   <statement resulttag="result">
      <sql>
       SELECT X4USER, STARTTIME FROM HISTORY WHERE PID='1279017294476'
      </sql>
   </statement>
</Root>
```

Der *JDBC Connector* gibt daraufhin z. B. folgendes Ergebnis-XML-Dokument aus, welches das Ergebnis der SQL-Abfrage sowie die Metadaten der *Result Sets* enthält:

```
<Data>
   <result-meta columns="2">
      <Column name="X4USER" label="X4USER" type="VARCHAR"
         nullable="true" auto-increment="false" read-only="false"
         writable="false" display-size="255" case-sensitive="true"
         searchable="true" signed="false" />
      <Column name="STARTTIME" label="STARTTIME" type="TIMESTAMP"
         nullable="true" auto-increment="false" read-only="false"
         writable="false" display-size="26" case-sensitive="false"
         searchable="true" signed="false" />
   </result-meta>
   <result>
      <X4USER>1</X4USER>
      <STARTTIME>2010-07-13 12:34:54.553</STARTTIME>
   </result>
</Data>
```

10.6.2 JDBC Iterator

`JDBC Iterator`: Stellt eine Verbindung zu einer JDBC-kompatiblen Datenbank her und führt ein oder mehrere Datenbank-Statements aus, die als Input-XML-Dokument mit adapterspezifischen Strukturen übergeben wurden. Der Iterator ruft die Daten blockweise ab und stellt diese auch blockweise zur Verarbeitung bereit. Dabei bleibt die Datenbankverbindung solange offen, bis alle Daten abgerufen wurden.

> Voraussetzung ist eine im Applikations-Server bereitgestellte JDBC-DataSource, für welche die Verbindungs-URL, Zugangsdaten, Schema sowie Transaktionsverhaltensweisen etc. definiert wurden. Nähere Informationen zum Bereitstellen von JDBC-DataSources entnehmen Sie der Dokumentation Ihres Applikations-Servers.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`Iterate`*: Datenmenge abhängig von dem im Parameter `fetchSize` angegebenen Wert aus der Datenbank auslesen und in einem Ergebnis-Dokument ausgeben • *`Clean`*: Iterator eines *JDBC Iterators* zurücksetzen, der dieselbe ID besitzt wie der gewünschte *JDBC Iterator* mit der Operation *`Iterate`* (siehe Parameter `id`)

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.jdbc.JdbcIterator*: Hauptklasse (Standard)
JndiName	JNDI-Name der DataSource Wird die Verbindung über den JNDI-Namen hergestellt, darf der Parameter `connectionURL` nicht ausgefüllt werden. **Mögliche Werte:** Gültiger JNDI-Name (z. B. *java:/DefaultDS*)

ColumnCase	Steuert wie die Spaltennamen ins Ergebnis übernommen werden. Da Datenbanken unterschiedliches Verhalten betreffend der Groß-/Kleinschreibung von Spaltennamen haben, kann man hiermit die Weiterverarbeitung des Ergebnisses unabhängig vom Datenbankverhalten machen. **Mögliche Werte:** • *NO_CHANGE*: Ändert die Spaltennamen nicht, d.h. die Schreibweise im Ergebnis ist exakt wie von der Datenbank geliefert (Standard) • *TO_LOWER*: Ändert die Spaltennamen auf Kleinbuchstaben • *TO_UPPER*: Ändert die Spaltennamen auf Großbuchstaben ⓘ Spaltenname, die keine gültigen Elementnamen (`NCNAME`) darstellen, werden durch `_<number>` (e. g. `_2`) ersetzt, wo *<number>* der Index der Spalte ist.
connectionURL	Verbindungs-URL für die JDBC-Datenbank ⓘ Wird die Verbindung über die Verbindungs-URL hergestellt, darf der Parameter `JndiName` nicht ausgefüllt werden. **Mögliche Werte:** • Die Art des Aufbaus hängt von der Datenbank ab, zu der eine Verbindung aufgebaut werden soll. • Beispiel für sqlite: *jdbc:sqlite:C:/Temp/mydatabase.db*
user	Benutzername zur Authentifizierung bei Verwendung des Parameters `connectionURL`
password	Passwort zur Authentifizierung bei Verwendung des Parameters `connectionURL`
CommitMode	Steuert, welche Variante für Commits verwendet werden soll **Mögliche Werte:** • *NONE*: Keinen Commit durchführen • *AUTO*: Automatischer Commit gemäß JDBC-API (d. h. nach jedem Batch) (Standard) • *STATEMENT*: Explizites Commit nach jedem Statement (d. h. nach allen Batches für dieses Statement) • *DOCUMENT*: Am Ende des Dokuments (d. h. nach Ausführung aller Statements) • *MANAGED*: Adapter in Transaktionen nutzen (für extern verwaltete Transaktionen)

AddMetaDataForQueries	Steuert, ob Datenbank-Metadaten, siehe JDBC Metadata Explorer, zu jedem *Result Set* im Ergebnis-XML-Dokument des Adapters ausgegeben werden sollen, wenn der Adapter eine SQL-Abfrage ausführt (nicht der Fall bei `UPDATE`, `DELETE` etc.). Im Batch-Modus (`StatementBatch`, `PreparedStatementBatch` und `CallableStatementBatch`) ist dies ebenfalls nicht möglich. **Mögliche Werte:** • *ja*: Dem Ergebnisdokument Datenbank-Metadaten hinzufügen • *nein*: Ergebnisdokument ohne Datenbank-Metadaten ausgeben (Standard)
fetchSize	Durch die Eingabe einer positiven ganzen Zahl lässt sich die Anzahl der Ergebnisse pro Abfrage beschränken. Sind Daten vorhanden, wird der Status `1` geliefert, ist das Ergebnis leer, wird der Status `0` ausgegeben. Wenn die eingegebene Zahl kleiner `1` ist, werden alle Daten ausgegeben.
id	Eindeutige ID des *JDBC Iterator*-Bausteins innerhalb des Prozess-Kontextes bzw. der Prozess-Instanz (wenn Sie mehrere voneinander unabhängig iterierende *JDBC Iteratoren* in demselben Prozess-Kontext verwenden möchten); **Mögliche Werte:** • *0* : Aktuelle Action-ID (Nummer des Prozess-Schritts des *JDBC Iterators* im aktuellen Prozess) als ID verwenden; Beachten Sie hierbei, dass ein *JDBC Iterator* in einem Subprozess die gleiche Action-ID besitzen könnte (Standard) • Beliebige ganze Zahl außer *0*, um den jeweiligen *JDBC Iterator*-Baustein innerhalb des Prozess-Kontextes eindeutig zu kennzeichnen; Empfehlung: einen Wert *>1000* verwenden, um nicht zufällig eine bereits bestehende Action-ID zu verwenden.

Statuswerte

1	Die Abfrage hat Datensätze geliefert bzw. die Anweisung hat Datensätze geändert Der Status 1 wird geliefert, wenn mehrere Abfragen bzw. Anweisungen im Dokument enthalten sind und mindestens eine Abfrage ein Ergebnis geliefert oder verändert hat. Der Adapter gibt diesen Status solange zurück, solange noch Datenblöcke zur Verarbeitung vorhanden sind.

0	Die Abfrage hat keine Ergebnisse geliefert oder die Anweisung hat keine Datensätze geändert Der Status `0` wird geliefert, wenn mehrere Abfragen bzw. Anweisungen im Dokument enthalten sind und keine Abfrage ein Ergebnis geliefert oder verändert hat.
-1	Technischer Fehler: genauere Informationen zur Fehlerursache entnehmen Sie dem Server-Log

Input

Der *JDBC Iterator* erwartet spezifische Input-XML-Strukturen, die Datenbank-Statements enthalten. Diese Input-XML-Strukturen können über XSL-Mappings in Abhängigkeit von den verarbeiteten XML-Daten dynamisch generiert werden.

10.6.2.1 Statement: Direktes SQL

Input

```
<Beliebiger Elementname>
   <statement resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
         SQL-Anweisungen
      </sql>
   </statement>
</Beliebiger Elementname>
```

Für jedes Element `<statement/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.
- Es enthält mindestens ein Element `<sql/>`, das jeweils eine SQL-Anweisung enthält.

> Falls Sie innerhalb der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA* -Block, z. B. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]>`

Beispiel

```
<select>
   <statement resulttag="result" timezone="UTC" format="dd.MM.yyyy">
      <sql>
         <![CDATA[SELECT * FROM JMS_USERS]]>
      </sql>
   </statement>
</select>
```

Output

Wurde die SQL-Anweisung erfolgreich ausgeführt, gibt der Datenbankverbindungs-Baustein ein XML-Dokument mit folgender Struktur zurück, das die Antwort der Datenbank als Zeichenkette (String) enthält:

```
<Data>
   <Ergebniselementname>
      <Feld1><!--Wert--></Feld1>
      <Feldn><!--Wert--></Feldn>
   </Ergebniselementname>
</Data>
```

Bei `SELECT`-Statements zum Zugriff auf Datenbanktabellen mit Spaltennamen, die keine gültigen XML-Namen darstellen, werden im Ergebnis-XML-Dokument synthetische Elementnamen erzeugt, um die Wohlgeformheit des XML-Dokuments sicherzustellen.

10.6.2.2 StatementBatch: Direktes SQL im Batch-Modus

Input

Der *JDBC Iterator* führt mehrere SQL-Anweisungen auf einmal im Batch-Modus aus, wenn sie in folgender Input-XML-Struktur abgelegt werden. Dabei sind lediglich SQL-Anweisungen möglich, die kein `ResultSet` zurückliefern, also z. B. keine Abfragen mit `SELECT`:

```
<Beliebiger Elementname>
   <statementbatch resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
         SQL-Anweisung 1
      </sql>
      <sql>
         SQL-Anweisung n
      </sql>
   </statementbatch>
</Beliebiger Elementname>
```

Für jedes Element `<statementbatch/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.
- Es enthält mindestens ein Element `<sql/>`, das jeweils eine SQL-Anweisung enthält.

> Falls Sie innerhalb der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=100 ]]>`

Beispiel

```
<select>
   <statementbatch resulttag="Ergebniselementname" timezone="UTC" format="dd.MM.yyyy">
      <sql>
         <![UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=100]]>
      </sql>
      <sql>
         <![UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=200]]>
      </sql>
   </statementbatch>
</select>
```

Output

Wurde die Batch-Anweisung erfolgreich ausgeführt, gibt der Datenbankverbindungs-Baustein folgende XML-Struktur zurück:

```
<Data>
   <Ergebniselementname>
      <!--Status der Batch-Operation-->
   </Ergebniselementname>
</Data>
```

Als Status der Batch-Operation sind folgende Ausgaben möglich:

- *(Anzahl der Zeilen)* `rows affected.` oder
- `SUCCESS_NO_INFO` oder
- `EXECUTE_FAILED`

10.6.2.3 PreparedStatement: SQL mit Werten

Der *JDBC Iterator* führt SQL-Anweisungen mit dynamischen Input-Daten aus, wenn sie in folgender Input-XML-Struktur abgelegt werden.

Alternativ können Sie ab Release 5.0 der X4 Suite auch SQL-Anweisungen mit benannten Parametern ausführen, siehe NamedPreparedStatement: SQL mit benannten Parametern.

Input

```
<Beliebiger Elementname>
   <preparedstatement resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
         <!--SQL-Anweisungen-->
      </sql>
      <columns>
         <column mode="in" name="frei wählbarer NCName"
          index="Platzhalter-Index" type="Datentyp"
          timezone="Zeitzone" format="Datumsformat">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--Wert--></Name>
         </row>
      </rowset>
   </preparedstatement>
</Beliebiger Elementname>
```

Für jedes Element `<preparedstatement/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß *java.text.SimpleDateFormat* an.
- Es enthält mindestens ein Element `<sql/>`.

Falls Sie innerhalb der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]>`

Für jedes Element `<column/>` gilt:

- Es darf beliebig häufig verwendet werden (*0* – *n*-mal)
- Das optionale Attribut `mode` gibt die "Richtung" des Datenstroms an (*in*: Input-Parameter für die Datenbank; *out*: Ergebnisparameter aus der Datenbank; *inout*: sowohl Input-Parameter als auch Ergebnisparameter; Standard: *in*.
- Das Attribut `name` gibt einen frei wählbaren Namen (*NCName*) für ein Parameter an.
- Das Attribut `index` gibt einen ganzzahligen Platzhalterindex für den in Attribut `name` definierten Parameter an.
- Das optionale Attribut `type` gibt den Datentyp an (z. B. *INTEGER* oder *BLOB*; Standard: *VARCHAR*)
  .

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *`GMT+2`*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.

Für jedes Element `<rowset/>` gilt:

- Es darf beliebig häufig verwendet werden (*`0`* – *`n`*–`mal`)
- Es enthält mindestens ein Element `<row/>`, das die Statement-Ausführung steuert.

Für jedes Element `<row/>` gilt:

- Es darf beliebig viele Elemente enthalten, die den Namen der Tabellenspalte oder den im SQL-Statement gesetzten Spalten-Alias tragen.
- Die darin liegenden Elemente enthalten die jeweiligen Werte.

Beachten Sie Folgendes

- Die Attribute `mode` und `format` sind in *X4 Server* Revision 31469 Pflichtparameter; ab neueren Versionen sind diese optional.
- Zulässige Datentypen für das Attribut `type` zum Setzen (Input-Parameter) und Lesen (*ResultSet*) von Werten sind: *`BIGINT`*, *`BINARY`*, *`BIT`*, *`BOOLEAN`*, *`BLOB`*, *`CHAR`*, *`CLOB`*, *`DATE`*, *`DECIMAL`*, *`DOUBLE`*, *`FLOAT`*, *`INTEGER`*, *`LONGNVARCHAR`*, *`LONGVARBINARY`*, *`LONGVARCHAR`*, *`NCHAR`*, *`NCLOB`*, *`NULL`*, *`NUMERIC`*, *`NVARCHAR`*, *`REAL`*, *`SMALLINT`*, *`SQLXML`*, *`TIME`*, *`TIMESTAMP`*, *`TINYINT`*, *`VARBINARY`* und *`VARCHAR`* (Standard), *`XML`* (alias für SQLXML).

Beispiel

```
<Rootelement>
   <preparedstatement resulttag="abc" timezone="MEZ"
      format="dd.MM.yyyy">
      <sql>
      <![CDATA[SELECT * FROM JMS_USERS WHERE USERID=?]]>
      </sql>
      <columns>
         <column mode="in" name="id" index="1"
            type="INTEGER">
         </column>
      </columns>
      <rowset>
         <row>
            <id>23</id>
         </row>
      </rowset>
   </preparedstatement>
</Rootelement>
```

Output

Wurde die SQL-Anweisung erfolgreich ausgeführt, gibt der Adapter ein XML-Dokument mit folgender Struktur zurück, das die Antwort der Datenbank als Zeichenkette (String) enthält:

```
<Data>
   <Ergebniselementname>
      <Feld1><!--Wert--></Feld1>
      <Feldn><!--Wert--></Feldn>
   </Ergebniselementname>
</Data>
```

10.6.2.4 PreparedStatementBatch: SQL mit Werten im Batch-Modus

Der *JDBC Iterator* führt mehrere SQL-Anweisungen auf einmal im Batch-Modus aus, wenn sie in folgender Input-XML-Struktur abgelegt werden. Dabei sind lediglich SQL-Anweisungen möglich, die kein Result Set zurückliefern, also z.B. keine Abfragen mit `SELECT`.

> Alternativ können Sie ab Release 5.0 der X4 Suite auch SQL-Anweisungen mit benannten Parametern mit oder ohne Batch-Modus ausführen, siehe NamedPreparedStatementBatch: SQL mit benannten Parametern im Batch-Modus.

Input

```
<Beliebiger Elementname>
   <preparedstatementbatch resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
         <!-- SQL-Anweisung -->
      </sql>
      <columns>
         <column mode="in" name="frei wählbarer NCName"
          index="Platzhalter-Index" type="Datentyp"
          timezone="Zeitzone" format="Datumsformat">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--Wert--></Name>
         </row>
         <!-- Weitere <row>-Elemente für Batch-Modus -->
      </rowset>
   </preparedstatementbatch>
</Beliebiger Elementname>
```

Für jedes Element `<preparedstatementbatch/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß *java.text.SimpleDateFormat* an.

- Es enthält mindestens ein Element `<sql/>`.

> Falls Sie innerhalb der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=? ]]>`

Für jedes Element `<column/>` gilt:

- Es darf beliebig häufig verwendet werden (*0 – n*-mal)
- Das optionale Attribut `mode` gibt die "Richtung" des Datenstroms an (*in*: Input-Parameter für die Datenbank; *out*: Ergebnisparameter aus der Datenbank; *inout*: sowohl Input-Parameter als auch Ergebnisparameter; Standard: *in*.
- Das Attribut `name` gibt einen frei wählbaren Namen (*NCName*) für ein Parameter an.
- Das Attribut `index` gibt einen ganzzahligen Platzhalterindex für den in Attribut `name` definierten Parameter an.
- Das optionale Attribut `type` gibt den Datentyp an (z. B. *INTEGER* oder *BLOB*; Standard: *VARCHAR*) .
- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.

Für jedes Element `<rowset/>` gilt:

- Es darf beliebig häufig verwendet werden (*0–n–mal*)
- Es enthält beliebig viele Elemente `<row/>`, welche die Statement-Ausführung steuern.

Für jedes Element `<row/>` gilt:

- Es darf beliebig viele Elemente enthalten, die den Namen der Tabellenspalte oder den im SQL-Statement gesetzten Spalten-Alias tragen.
- Die darin liegenden Elemente enthalten die jeweiligen Werte.

> **Beachten Sie Folgendes**
>
> - Die Attribute `mode` und `format` sind in *X4 Server* Revision 31469 Pflichtparameter; ab neueren Versionen sind diese optional.
> - Zulässige Datentypen für das Attribut `type` zum Setzen (Input-Parameter) und Lesen (*ResultSet*) von Werten sind: *BIGINT, BINARY, BIT, BOOLEAN, BLOB, CHAR, CLOB, DATE, DECIMAL, DOUBLE, FLOAT, INTEGER, LONGNVARCHAR, LONGVARBINARY, LONGVARCHAR, NCHAR, NCLOB, NULL, NUMERIC, NVARCHAR, REAL, SMALLINT, SQLXML, TIME, TIMESTAMP, TINYINT, VARBINARY und VARCHAR* (Standard), *XML* (alias für SQLXML).

Beispiel

```
<Rootelement>
   <preparedstatementbatch resulttag="Ergebniselementname" timezone="MEZ"
      format="dd.MM.yyyy">
      <sql>
      <![CDATA[ UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=? ]]>
      </sql>
      <columns>
         <column mode="in" name="id" index="1"
            type="INTEGER">
         </column>
      </columns>
      <rowset>
         <row>
            <id>23</id>
         </row>
         <row>
            <id>25</id>
         </row>
      </rowset>
   </preparedstatementbatch>
</Rootelement>
```

Output

Wurde die Batch-Anweisung erfolgreich ausgeführt, gibt der Datenbankverbindungs-Baustein folgende XML-Struktur zurück:

```
<Data>
   <Ergebniselementname>
      <!--Status der Batch-Operation-->
   </Ergebniselementname>
</Data>
```

Als Status der Batch-Operation sind folgende Ausgaben möglich:

- *(Anzahl der Zeilen)* `rows affected.` oder
- `SUCCESS_NO_INFO` oder
- `EXECUTE_FAILED`

10.6.2.5 CallableStatement: Stored Procedures

Der *JDBC Iterator* führt Stored Procedures über *JDBC Callablestatements* mit dynamischen Input-Daten aus, wenn sie in folgender Input-XML-Struktur abgelegt werden.

Input

```
<Beliebiger Elementname>
   <callablestatement resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
        <!--SQL-Anweisungen -->
      </sql>
      <columns>
         <column mode="Modus" name="frei wählbarer Name" index="Platzhalter"
          type="Datentyp" timezone="Zeitzone" format="Datumsformat">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--Wert--></Name>
         </row>
      </rowset>
   </callablestatement>
</Beliebiger Elementname>
```

Für jedes Element `<callablestatement/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.
- Es enthält mindestens ein Element `<sql/>`.

> Falls Sie innerhalb der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]>`

Für jedes Element `<column/>` gilt:

- Es darf beliebig häufig verwendet werden (*0–n*-mal)
- Das optionale Attribut `mode` gibt die "Richtung" des Datenstroms an (*in*: Input-Parameter für die Datenbank; *out*: Ergebnisparameter aus der Datenbank; *inout*: sowohl Input-Parameter als auch Ergebnisparameter; siehe http://download.oracle.com/javase/1,5.0/docs/guide/jdbc/getstart/callablestatement.html); Standard: *in*.
- Das Attribut `name` gibt einen frei wählbaren Namen für ein Parameter an.
- Das Attribut `index` gibt einen ganzzahligen Platzhalterindex für den in Attribut `name` definierten Parameter an.
- Das optionale Attribut `type` gibt den Datentyp an (zulässige Datentypen siehe Hinweis unten; Standard: *VARCHAR*).
- Das Attribut `name` gibt einen frei wählbaren Namen für ein Parameter an.
- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.

- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.

Für jedes Element `<rowset/>` gilt:

- Es darf beliebig häufig verwendet werden (*0–n*-mal).
- Es enthält beliebig viele Elemente `<row/>`, die die Statement-Ausführung steuern.

Für jedes Element `<row/>` gilt:

- Es darf beliebig viele Elemente enthalten, die den Namen der Tabellenspalte oder den im SQL-Statement gesetzten Spalten-Alias tragen.
- Die darin liegenden Elemente enthalten die jeweiligen Werte.

Beachten Sie Folgendes

- Die Attribute `mode` und `format` sind in *X4 Server* Revision 31469 Pflichtparameter; ab neueren Versionen sind diese optional.
- Zulässige Datentypen für das Attribut `type` zum Setzen (Input-Parameter) und Lesen (*ResultSet*) von Werten sind: *`BIGINT`, `BINARY`, `BIT`, `BOOLEAN`, `BLOB`, `CHAR`, `CLOB`, `DATE`, `DECIMAL`, `DOUBLE`, `FLOAT`, `INTEGER`, `LONGNVARCHAR`, `LONGVARBINARY`, `LONGVARCHAR`, `NCHAR`, `NCLOB`, `NULL`, `NUMERIC`, `NVARCHAR`, `REAL`, `SMALLINT`, `SQLXML`, `TIME`, `TIMESTAMP`, `TINYINT`, `VARBINARY` und `VARCHAR`* (Standard), *`XML`* (alias für SQLXML).

Beispiel

```
<Rootelement>
   <callablestatement resulttag="abc" timezone="MEZ"
      format="dd.MM.yyyy">
      <sql>
      <![CDATA[ {call getData(?)} ]]>
      </sql>
      <columns>
         <column mode="out" name="id" index="1"
            type="INTEGER">
         </column>
      </columns>
      <rowset>
         <row>
            <id>TestUser</id>
         </row>
      </rowset>
   </callablestatement>
</Rootelement>
```

Output

Wurde die SQL-Anweisung erfolgreich ausgeführt, gibt der Datenbankverbindungs-Baustein ein XML-Dokument mit folgender Struktur zurück, das die Antwort der Datenbank als Zeichenkette (String) enthält:

```
<Data>
   <Ergebniselementname>
      <Feld1><!--Wert--></Feld1>
      <Feldn><!--Wert--></Feldn>
   </Ergebniselementname>
</Data>
```

10.6.2.6 CallableStatementBatch: Stored Procedures im Batch-Modus

Der *JDBC Iterator* führt Stored Procedures über *JDBC Callablestatements* mit dynamischen Input-Daten aus, wenn sie in folgender Input-XML-Struktur abgelegt werden. Dabei sind lediglich SQL-Anweisungen möglich, die kein ResultSet zurückliefern, also z. B. keine Abfragen mit `SELECT`.

Input

```
<Beliebiger Elementname>
   <callablestatementbatch resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
        <!--SQL-Anweisungen -->
      </sql>
      <columns>
         <column mode="Modus" name="frei wählbarer Name" index="Platzhalter"
          type="Datentyp" timezone="Zeitzone" format="Datumsformat">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--Wert--></Name>
         </row>
      </rowset>
   </callablestatementbatch>
</Beliebiger Elementname>
```

Für jedes Element `<callablestatementbatch/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.
- Es enthält mindestens ein Element `<sql/>`.

Falls Sie innerhalb der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]`

Für jedes Element `<column/>` gilt:

- Es darf beliebig häufig verwendet werden (*0* – *n*-mal)
- Das optionale Attribut `mode` gibt die "Richtung" des Datenstroms an (*in*: Input-Parameter für die Datenbank; *out*: Ergebnisparameter aus der Datenbank; *inout*: sowohl Input-Parameter als auch Ergebnisparameter; siehe http://download.oracle.com/javase/1,5.0/docs/guide/jdbc/getstart/callablestatement.html); Standard: *in*.
- Das Attribut `name` gibt einen frei wählbaren Namen für ein Parameter an.
- Das Attribut `index` gibt einen ganzzahligen Platzhalterindex für den in Attribut `name` definierten Parameter an.
- Das optionale Attribut `type` gibt den Datentyp an (zulässige Datentypen siehe Hinweis unten; Standard: *VARCHAR*).
- Das Attribut `name` gibt einen frei wählbaren Namen für ein Parameter an.
- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an.

Für jedes Element `<rowset/>` gilt:

- Es darf beliebig häufig verwendet werden (*0* – *n*-mal).
- Es enthält beliebig viele Elemente `<row/>`, die die Statement-Ausführung steuern.

Für jedes Element `<row/>` gilt:

- Es darf beliebig viele Elemente enthalten, die den Namen der Tabellenspalte oder den im SQL-Statement gesetzten Spalten-Alias tragen.
- Die darin liegenden Elemente enthalten die jeweiligen Werte.

Beachten Sie Folgendes

- Die Attribute `mode` und `format` sind in *X4 Server* Revision 31469 Pflichtparameter; ab neueren Versionen sind diese optional.
- Zulässige Datentypen für das Attribut `type` zum Setzen (Input-Parameter) und Lesen (*ResultSet*) von Werten sind: *BIGINT, BINARY, BIT, BOOLEAN, BLOB, CHAR, CLOB, DATE, DECIMAL, DOUBLE, FLOAT, INTEGER, LONGNVARCHAR, LONGVARBINARY, LONGVARCHAR, NCHAR, NCLOB, NULL, NUMERIC, NVARCHAR, REAL, SMALLINT, SQLXML, TIME, TIMESTAMP, TINYINT, VARBINARY und VARCHAR* (Standard), *XML* (alias für SQLXML).

Beispiel

```
<Rootelement>
   <callablestatementbatch resulttag="abc" timezone="MEZ"
      format="dd.MM.yyyy">
      <sql>
      <![CDATA[ {call getData(?)} ]]>
      </sql>
      <columns>
         <column mode="out" name="id" index="1"
            type="INTEGER">
         </column>
      </columns>
      <rowset>
         <row>
            <id>TestUser</id>
         </row>
      </rowset>
   </callablestatementbatch>
   <callablestatementbatch resulttag="abc" timezone="MEZ"
      format="dd.MM.yyyy">
      <sql>
      <![CDATA[ {call getData(?)} ]]>
      </sql>
      <columns>
         <column mode="out" name="id" index="1"
            type="INTEGER">
         </column>
      </columns>
      <rowset>
         <row>
            <id>TestUser</id>
         </row>
      </rowset>
   </callablestatementbatch>
</Rootelement>
```

Output

Wurde die Batch-Anweisung erfolgreich ausgeführt, gibt der Datenbankverbindungs-Baustein folgende XML-Struktur zurück:

```
<Data>
   <Ergebniselementname>
      <!--Status der Batch-Operation-->
   </Ergebniselementname>
</Data>
```

Als Status der Batch-Operation sind folgende Ausgaben möglich:

- *(Anzahl der Zeilen)* `rows affected.` oder
- `SUCCESS_NO_INFO` oder
- `EXECUTE_FAILED`

10.6.2.7 NamedPreparedStatement: SQL mit benannten Parametern

Der *JDBC Iterator* führt SQL-Anweisungen mit benannten Parametern aus, wenn sie in folgender Input-XML-Struktur abgelegt werden.

Input

```
<Beliebiger Elementname>
   <namedpreparedstatement resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
         <!--SQL-Anweisungen (optional innerhalb eines CDATA-Blocks) -->
         <!--Je Parameter ein XML-Element, gleichnamig zum entsprechenden Kindelement
in <row> -->
      </sql>
      <rowset>
         <row>
            <Name><!--Wert--></Name>
         </row>
      </rowset>
   </namedpreparedstatement>
</Beliebiger Elementname>
```

Für jedes Element `<namedpreparedstatement/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß *java.text.SimpleDateFormat* an und bestimmt das Format der zurückgegebenen Datumswerten.
- Es enthält ein Element `<sql/>`.

Für jedes XML-Element eines benannten Parameters innerhalb der SQL-Anweisung in Element `<sql/>` gilt:

- Der Elementname gibt den Parameter-Namen an, der mit entsprechenden Element innerhalb von `<row/>` übereinstimmt.
- Das optionale Attribut `mode` gibt die "Richtung" des Datenstroms an (*in*: Input-Parameter für die Datenbank; *out*: Ergebnisparameter aus der Datenbank; *inout*: sowohl Input-Parameter als auch Ergebnisparameter; Standard: *in*.
- Das optionale Attribut `type` gibt den Datentyp an, z. B. *INTEGER*, *BLOB* oder *VARCHAR*(Standard).
- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an und gibt das erwartete Format für den Wert innerhalb des entsprechenden `<row>`-Elements vor.

Falls Sie in der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ SELECT * FROM TABLE WHERE ID<]]><id type="INTEGER" />`.

Für jedes Element `<rowset/>` gilt:

- Es darf beliebig häufig verwendet werden (*0 – n*–mal)
- Es enthält beliebig viele Elemente `<row/>`, die die Statement-Ausführung steuern.

Für jedes Element `<row/>` gilt:

- Es muss mindestens ein Element enthalten, das dem Elementnamen im SQL-Statement entspricht.
- Die darin liegenden Elemente enthalten die jeweiligen Werte.

Beachten Sie für jedes Datenelement innerhalb von `<row/>`, dass im Elementinhalt keine unbeabsichtigten Zeilenumbrüche definiert wurden (z.B. durch Editor-Formatierung). Andernfalls würden diese Zeilenumbrüche in der erzeugten SQL-Anweisung als `\n` angefügt.

Beachten Sie Folgendes
Zulässige Datentypen für das Attribut `type` zum Setzen (Input-Parameter) und Lesen (*ResultSet*) von Werten sind: *BIGINT, BINARY, BIT, BOOLEAN, BLOB, CHAR, CLOB, DATE, DECIMAL, DOUBLE, FLOAT, INTEGER, LONGNVARCHAR, LONGVARBINARY, LONGVARCHAR, NCHAR, NCLOB, NULL, NUMERIC, NVARCHAR, REAL, SMALLINT, SQLXML, TIME, TIMESTAMP, TINYINT, VARBINARY und VARCHAR* (Standard), *XML* (alias für SQLXML).

Beispiel

```
<Rootelement>
    <namedpreparedstatement resulttag="namedPreparedStatementWithCData"
     timezone="UTC" format="dd.MM.yyyy">
        <sql>
             SELECT * FROM PROCESS WHERE X4_DURATION= <duration type="INTEGER"
                mode="in" /> AND X4_PROCESSNAME= <processname
                type="VARCHAR" mode="in" />
        </sql>
        <rowset>
            <row>
                <duration>101</duration>
                <processname>Project/Process/Path/Process.wrf</processname>
            </row>
        </rowset>
    </namedpreparedstatement>
</Rootelement>
```

Output

Wurde die SQL-Anweisung erfolgreich ausgeführt, gibt der Adapter ein XML-Dokument mit folgender Struktur zurück, das die Antwort der Datenbank als Zeichenkette (String) enthält:

```
<Data>
   <Ergebniselementname>
      <Feld1><!--Wert--></Feld1>
      <Feldn><!--Wert--></Feldn>
   </Ergebniselementname>
</Data>
```

10.6.2.8 NamedPreparedStatementBatch: SQL mit benannten Parametern im Batch-Modus

Der *JDBC Iterator* führt mehrere SQL-Anweisungen auf einmal im Batch-Modus aus, wenn sie in folgender Input-XML-Struktur jeweils in einem `<namedpreparedstatementbatch>`-Block abgelegt werden. Dabei sind lediglich SQL-Anweisungen möglich, die kein ResultSet zurückliefern, also z. B. keine Abfragen mit `SELECT`.

Input

```
<Beliebiger Elementname>
   <namedpreparedstatementbatch resulttag="Ergebniselementname" timezone="Zeitzone"
    format="Datumsformat">
      <sql>
         <!--SQL-Anweisungen (optional innerhalb eines CDATA-Blocks) -->
         <!--Je Parameter ein XML-Element, gleichnamig zum entsprechenden Kindelement
in <row> -->
      </sql>
      <rowset>
         <row>
            <Name><!--Wert--></Name>
         </row>
         <!-- Weitere <row>-Elemente für Batch-Modus -->
      </rowset>
   </namedpreparedstatementbatch>
</Beliebiger Elementname>
```

Für jedes Element `<namedpreparedstatementbatch/>` gilt:

- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *GMT+2*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß *java.text.SimpleDateFormat* an und bestimmt das Format der zurückgegebenen Datumswerten.
- Es enthält mindestens ein Element `<sql/>`.

Für jedes XML-Element eines benannten Parameters innerhalb der SQL-Anweisung im Element `<sql/>` gilt:

- Der Elementname gibt den Parameter-Namen an, der mit entsprechenden Element innerhalb von `<row/>` übereinstimmt.
- Das optionale Attribut `mode` gibt die "Richtung" des Datenstroms an (*`in`*: Input-Parameter für die Datenbank; *`out`*: Ergebnisparameter aus der Datenbank; *`inout`*: sowohl Input-Parameter als auch Ergebnisparameter; Standard: *`in`*.
- Das optionale Attribut `type` gibt den Datentyp an, z. B. *`INTEGER`*, *`BLOB`* oder *`VARCHAR`* (Standard).
- Das optionale Attribut `timezone` gibt eine Zeitzone gemäß `java.util.TimeZone`, siehe http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html, an, z. B. *`GMT+2`*.
- Das optionale Attribut `format` gibt ein Datumsformat gemäß `java.text.SimpleDateFormat` an und gibt das erwartete Format für den Wert innerhalb des entsprechenden `<row>`-Elements vor.

> Falls Sie in der SQL-Anweisung Zeichen verwenden, die in XML eine Bedeutung besitzen, maskieren Sie die SQL-Anweisung mit einem umgebenden *CDATA*-Block, z. B. `<![CDATA[ SELECT * FROM TABLE WHERE ID<]]><id type="INTEGER" />`.

Für jedes Element `<rowset/>` gilt:

- Es darf beliebig häufig verwendet werden (*`0`*–*`n`*–`mal`)
- Es enthält beliebig viele Elemente `<row/>`, die die Statement-Ausführung steuern.

Für jedes Element `<row/>` gilt:

- Es muss mindestens ein Element enthalten, das dem Elementnamen im SQL-Statement entspricht.
- Die darin liegenden Elemente enthalten die jeweiligen Werte.

Beachten Sie für jedes Datenelement innerhalb von `<row/>`, dass im Elementinhalt keine unbeabsichtigten Zeilenumbrüche definiert wurden (z.B. durch Editor-Formatierung). Andernfalls würden diese Zeilenumbrüche in der erzeugten SQL-Anweisung als `\n` angefügt.

> **Beachten Sie Folgendes**
> Zulässige Datentypen für das Attribut `type` zum Setzen (Input-Parameter) und Lesen (*ResultSet*) von Werten sind: *`BIGINT`*, *`BINARY`*, *`BIT`*, *`BOOLEAN`*, *`BLOB`*, *`CHAR`*, *`CLOB`*, *`DATE`*, *`DECIMAL`*, *`DOUBLE`*, *`FLOAT`*, *`INTEGER`*, *`LONGNVARCHAR`*, *`LONGVARBINARY`*, *`LONGVARCHAR`*, *`NCHAR`*, *`NCLOB`*, *`NULL`*, *`NUMERIC`*, *`NVARCHAR`*, *`REAL`*, *`SMALLINT`*, *`SQLXML`*, *`TIME`*, *`TIMESTAMP`*, *`TINYINT`*, *`VARBINARY`* und *`VARCHAR`* (Standard), *`XML`* (alias für SQLXML).

Beispiel

```
<Rootelement>
    <namedpreparedstatementbatch resulttag="namedPreparedStatementBatchWithCData"
     timezone="UTC" format="dd.MM.yyyy">
        <sql>
             <![CDATA[UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=]]>
             <duration type="INTEGER" mode="inout" />
        </sql>
        <rowset>
            <row>
                <duration>101</duration>
            </row>
            <row>
                <duration>9</duration>
            </row>
        </rowset>
    </namedpreparedstatementbatch>
</Rootelement>
```

Output

Wurde die SQL-Anweisung erfolgreich ausgeführt, gibt der Adapter ein XML-Dokument mit folgender Struktur zurück:

```
<Data>
   <Ergebniselementname count="Anzahl der betroffenen Datenbankeinträge">
      <!--Status der Batch-Operation-->
   </Ergebniselementname>
</Data>
```

Als Status der Batch-Operation sind folgende Ausgaben möglich:

- *(Anzahl der Zeilen)* `rows affected.` oder
- `SUCCESS_NO_INFO` oder
- `EXECUTE_FAILED`

10.6.2.9 Abrufen von Datenbank-Metadaten

Mit dem Adapter-Parameter `AddMetaDataForQueries` (ab *X4 Server* Revision *32408*) können Sie steuern, ob Datenbank-Metadaten zu den entsprechenden *Result Sets* im Ergebnis-XML-Dokument innerhalb eines XML-Elements ermittelt und gemeinsam mit den Ergebnissen der SQL-Abfrage ausgegeben werden sollen.

Der Name jedes XML-Elements, das die entsprechenden Metadaten enthält, setzt sich aus dem Wert des Attributs `resulttag` im Input-Dokument und dem Zusatz `-meta` zusammen, z. B. `<result-meta>`.

Bei SQL-Statements, die kein *Result Set* ausgeben (z. B. bei `UPDATE` oder `DELETE`), können keine Metadaten ausgegeben werden. Im Batch-Modus des Adapters (`StatementBatch`, `PreparedStatementBatch` und `CallableStatementBatch`) ist dies ebenfalls nicht möglich.

Beispiel

Im folgenden Input-XML-Dokument wird eine SQL-Abfrage und als Ergebniselement der Name `result` definiert:

```
<Root>
   <statement resulttag="result">
      <sql>
       SELECT X4USER, STARTTIME FROM HISTORY WHERE PID='1279017294476'
      </sql>
   </statement>
</Root>
```

Der *JDBC Iterator* gibt daraufhin z. B. folgendes Ergebnis-XML-Dokument aus, welches das Ergebnis der SQL-Abfrage sowie die Metadaten der *Result Sets* enthält:

```
<Data>
   <result-meta columns="2">
      <Column name="X4USER" label="X4USER" type="VARCHAR"
         nullable="true" auto-increment="false" read-only="false"
         writable="false" display-size="255" case-sensitive="true"
         searchable="true" signed="false" />
      <Column name="STARTTIME" label="STARTTIME" type="TIMESTAMP"
         nullable="true" auto-increment="false" read-only="false"
         writable="false" display-size="26" case-sensitive="false"
         searchable="true" signed="false" />
   </result-meta>
   <result>
      <X4USER>1</X4USER>
      <STARTTIME>2010-07-13 12:34:54.553</STARTTIME>
   </result>
</Data>
```

10.6.3 JDBC Metadata Explorer

`JDBC Metadata Explorer`: Stellt eine Verbindung zu einer JDBC-kompatiblen Datenbank her und ermittelt Metadaten zur Datenbank, zu Tabellen und Spalten sowie zu hinterlegten Primär- und Fremdschlüsseln. Der Adapter erwartet ein Input-XML-Dokument mit adapterspezifischen Strukturen, um die gewünschten Metadaten abzufragen.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *execute*: Adapter ausführen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.jdbc.meta.MetadataExplorer*: Hauptklasse (Standard)
JndiName	JNDI-Name der DataSource Wird die Verbindung über den JNDI-Namen hergestellt, darf der Parameter `connectionURL` nicht ausgefüllt werden. **Mögliche Werte:** Gültiger JNDI-Name (z.B. *java:/DefaultDS*)
ColumnCase	Steuert wie die Spaltennamen ins Ergebnis übernommen werden. Da Datenbanken unterschiedliches Verhalten betreffend der Groß-/Kleinschreibung von Spaltennamen haben, kann man hiermit die Weiterverarbeitung des Ergebnis unabhängig vom Datenbankverhalten machen. **Mögliche Werte:** • *NO_CHANGE*: Ändert die Spaltennamen nicht, d.h. die Schreibweise im Ergebnis ist exakt wie von der Datenbank geliefert (Standard) • *TO_LOWER*: Ändert die Spaltennamen auf Kleinbuchstaben • *TO_UPPER*: Ändert die Spaltennamen auf Großbuchstaben Spaltenname, die keine gültigen Elementnamen (`NCNAME`) darstellen, werden durch `_<number>` (z.B. `_2`) ersetzt, wobei *<number>* der Index der Spalte ist.
connectionURL	Verbindungs-URL für die JDBC-Datenbank Wird die Verbindung über die Verbindungs-URL hergestellt, darf der Parameter `JndiName` nicht ausgefüllt werden. **Mögliche Werte:** • Die Art des Aufbaus hängt von der Datenbank ab, zu der eine Verbindung aufgebaut werden soll. • Beispiel für sqlite: *jdbc:sqlite:C:/Temp/mydatabase.db*
user	Benutzername zur Authentifizierung bei Verwendung des Parameters `connectionURL`

password	Passwort zur Authentifizierung bei Verwendung des Parameters `connectionURL`

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Der *JDBC Metadata Explorer* erwartet spezifische Input-XML-Strukturen, die je nach Bedarf eine oder mehrere der im Folgenden beschriebenen Metadaten-Abfragen enthalten können. Diese Input-XML-Strukturen können über XSL-Mappings in Abhängigkeit von den verarbeiteten XML-Daten dynamisch generiert werden.

- Datenbank-Metadaten abrufen
- Tabellen-Metadaten abrufen
- Spalten-Metadaten abrufen
- Primärschlüssel einer Tabelle abrufen
- Fremdschlüssel einer Tabelle abrufen
- Fremdschlüssel-Referenzen abrufen

10.6.3.1 Datenbank-Metadaten abrufen

Mit dem *JDBC Metadata Explorer* können Sie verschiedene Metadaten (Hersteller-/Produktname, unterstützte Funktionen etc.) und die vorhandenen Schemas von der via JDBC verbundenen Datenbank abrufen.

Input

Einem Input-XML-Dokument fügen Sie dazu innerhalb eines beliebigen Wurzel-Elements das Element `<databasemeta resulttag=" Ergebniselementname "/>` hinzu, in dessen Attribut `resulttag` Sie einen gültigen XML-Namen angeben.

```
<Root>
  <databasemeta resulttag="Ergebniselementname"/>
</Root>
```

Ausgabe

Die XML-Struktur, die vom Adapter ausgegeben wird, besitzt innerhalb des Wurzel-Elements `<Data>` ein Element, dessen Name in der Input-Struktur im Attribut `resulttag` angegeben wurde und das die Schema-Informationen und Datenbank-Metadaten enthält:

```
<Data>
   <Ergebniselementname eigenschaft="wert" ...>
   <Schema>
      <TABLE_SCHEM>DB-Schema</TABLE_SCHEM>
      <TABLE_CATALOG isNull="(true/false)" />
   </Schema>
   ...
   </Ergebniselementname>
</Data>
```

Beispiel

In einem Input-XML-Dokument wird folgende XML-Struktur definiert, um Metadaten zur verbundenen Datenbank auszugeben:

```
<RootElement>
   <databasemeta resulttag="DB"/>
</RootElement>
```

Der *JDBC Metadata Explorer* gibt daraufhin z. B. folgendes Ergebnis-XML-Dokument aus:

```
<Data>
   <DB product-name="Apache Derby" readOnly="false"
      supports-stored-procedures="true" supports-batch-updates="true"
      all-tables-are-selectable="true" supports-outer-joins="true"
      supports-convert="true" supports-group-by="true">
      <Schema>
         <TABLE_SCHEM>APP</TABLE_SCHEM>
         <TABLE_CATALOG isNull="true" />
      </Schema>
      ...
   </DB>
```

10.6.3.2 Tabellen-Metadaten abrufen

Mit dem *JDBC Metadata Explorer* können Sie Tabellen-Metadaten (Name, Typ etc.) zu einer oder mehreren Datenbank-Tabellen abrufen; hierbei können Sie die Suche nach Katalog, Schema und Tabellenname eingrenzen.

Input

Einem Input-XML-Dokument fügen Sie dazu innerhalb eines beliebigen Wurzel-Elements das Element `<tablemetadata resulttag="` *`Ergebniselementname`* `"/>` hinzu, in dessen Attribut `resulttag` Sie einen gültigen XML-Namen angeben. Die Angabe von Katalog, Schema und Tabelle in den

Unterelementen <catalog>, <schema> und <table> sind hierbei optional; wenn diese Such-Einschränkungen fehlen, werden sämtliche verfügbaren Metadaten abgerufen.

```
<Root>
  <tablemetadata resulttag="Ergebniselementname">
     <catalog>Katalog</catalog>
     <schema>DB-Schema</schema>
     <table>DB-Tabelle</table>
  </tablemetadata>
</Root>
```

Ausgabe

Die XML-Struktur, die vom Adapter ausgegeben wird, besitzt innerhalb des Wurzel-Elements <Data> ein XML-Element, dessen Elementname in der Input-Struktur im Attribut `resulttag` angegeben wurde und das die Tabellen-Metadaten enthält. Wenn es zu einem Abruf mehrere Ergebnisse gibt (d. h. mehrere Tabellen oder Sichten), so werden diese jeweils in einem separaten Element ausgegeben.

```
<Data>
   <Ergebniselementname>
      <Tabellen-Eigenschaft>Wert</Tabellen-Eigenschaft>
      ...
   </Ergebniselementname>
</Data>
```

ⓘ Erläuterungen (englischsprachig) zu den einzelnen Tabellen-Eigenschaften finden Sie hier.

Beispiel

In einem Input-XML-Dokument wird folgende XML-Struktur definiert, um Metadaten zu einer Tabelle `HISTORY` im Datenbankschema APP auszugeben:

```
<Root>
  <tablemetadata resulttag="DBTable">
     <catalog></catalog>
     <schema>APP</schema>
     <table>HISTORY</table>
  </tablemetadata>
</Root>
```

Der *JDBC Metadata Explorer* gibt daraufhin z. B. folgendes Ergebnis-XML-Dokument aus:

```
<Data>
   <DBTable>
      <TABLE_CAT />
      <TABLE_SCHEM>APP</TABLE_SCHEM>
      <TABLE_NAME>HISTORY</TABLE_NAME>
      <TABLE_TYPE>TABLE</TABLE_TYPE>
      <REMARKS />
      <TYPE_CAT isNull="true" />
      <TYPE_SCHEM isNull="true" />
      <TYPE_NAME isNull="true" />
      <SELF_REFERENCING_COL_NAME isNull="true" />
      <REF_GENERATION isNull="true" />
   </DBTable>
</Data>
```

10.6.3.3 Spalten-Metadaten abrufen

Mit dem *JDBC Metadata Explorer* können Sie Spalten-Metadaten (Datentyp, Nullable, Autoincrement etc.) zu einer oder mehreren Tabellenspalten abrufen; hierbei können Sie die Suche nach Katalog, Schema, Tabelle und Spalte eingrenzen.

Input

Einem Input-XML-Dokument fügen Sie dazu innerhalb eines beliebigen Wurzel-Elements das Element `<tablecolumns resulttag=" Ergebniselementname "/>` hinzu, in dessen Attribut `resulttag` Sie einen gültigen XML-Namen angeben. Die Angabe von Katalog, Schema, Tabelle und Spalte in den Unterelementen `<catalog>`, `<schema>`, `<table>` und `<column>` sind hierbei optional; wenn diese Such-Einschränkungen fehlen, werden sämtliche verfügbaren Metadaten abgerufen.

```
<Root>
  <tablemetadata resulttag="Ergebniselementname">
     <catalog>Katalog</catalog>
     <schema>DB-Schema</schema>
     <table>DB-Tabelle</table>
     <column>Spalte</column>
  </tablemetadata>
</Root>
```

Ausgabe

Die XML-Struktur, die vom Adapter ausgegeben wird, besitzt innerhalb des Wurzel-Elements `<Data>` ein Element, dessen Name in der Input-Struktur im Attribut `resulttag` angegeben wurde und das die Spalten-Metadaten enthält. Wenn es mehrere Ergebnisse gibt (d. h. mehrere Spalten), so werden diese jeweils in einem separaten Element ausgegeben.

```
<Data>
   <Ergebniselementname>
      <Spalten-Eigenschaft>Wert</Spalten-Eigenschaft>
      ...
   </Ergebniselementname>
</Data>
```

Erläuterungen (englischsprachig) zu den einzelnen Spalten-Eigenschaften finden Sie hier.

Beispiel 1

In einem Input-XML-Dokument wird folgende XML-Struktur definiert, um Metadaten zu einer Spalte `X4_PID` in der Tabelle `HISTORY` im Datenbankschema APP auszugeben:

```
<Root>
  <tablecolumns resulttag="DBColumn">
     <catalog></catalog>
     <schema>APP</schema>
     <table>HISTORY</table>
     <column>X4_PID</column>
  </tablecolumns>
</Root>
```

Der *JDBC Metadata Explorer* gibt daraufhin z. B. folgendes Ergebnis-XML-Dokument aus:

```
<Data>
   <DBColumn>
      <TABLE_CAT />
      <TABLE_SCHEM>APP</TABLE_SCHEM>
      <TABLE_NAME>HISTORY</TABLE_NAME>
      <COLUMN_NAME>X4_PID</COLUMN_NAME>
      <DATA_TYPE>12</DATA_TYPE>
      <TYPE_NAME>VARCHAR</TYPE_NAME>
      <COLUMN_SIZE>20</COLUMN_SIZE>
      <BUFFER_LENGTH isNull="true" />
      <DECIMAL_DIGITS isNull="true" />
      <NUM_PREC_RADIX isNull="true" />
      <NULLABLE>0</NULLABLE>
      <REMARKS/>
      <COLUMN_DEF isNull="true" />
      <SQL_DATA_TYPE isNull="true" />
      <SQL_DATETIME_SUB isNull="true" />
      <CHAR_OCTET_LENGTH>40</CHAR_OCTET_LENGTH>
      <ORDINAL_POSITION>1</ORDINAL_POSITION>
      <IS_NULLABLE>NO</IS_NULLABLE>
      <SCOPE_CATLOG isNull="true" />
      <SCOPE_SCHEMA isNull="true" />
      <SCOPE_TABLE isNull="true" />
      <SOURCE_DATA_TYPE isNull="true" />
      <IS_AUTOINCREMENT>NO</IS_AUTOINCREMENT>
   </DBColumn>
</Data>
```

Beispiel 2

In einem Input-XML-Dokument wird folgende XML-Struktur definiert, um Metadaten zu einer Spalte `X4_PID` in der Tabelle `HISTORY` im Datenbankschema APP auszugeben. Das Element `<column>` mit dem Attribut `label` innerhalb von `<resultsetRestriction>` schränkt dabei die ausgegebenen Daten ein:

```
<Root>
  <tablecolumns resulttag="DBColumn">
     <catalog></catalog>
     <schema>APP</schema>
     <table>HISTORY</table>
     <column>X4_PID</column>
     <resultsetRestriction>
        <column label="TABLE_NAME" />
        <column label="DATA_TYPE" />
        <column label="NULLABLE" />
        <column label="TYPE_NAME" />
     </resultsetRestriction>
    </tablecolumns>
</Root>
```

Der *JDBC Metadata Explorer* gibt daraufhin z. B. folgendes Ergebnis-XML-Dokument aus:

```
<Data>
   <DBColumn>
      <TABLE_NAME>HISTORY</TABLE_NAME>
      <COLUMN_NAME>X4_PID</COLUMN_NAME>
      <DATA_TYPE>12</DATA_TYPE>
      <TYPE_NAME>VARCHAR</TYPE_NAME>
   </DBColumn>
</Data>
```

10.6.3.4 Primärschlüssel einer Tabelle abrufen

Mit dem *JDBC Metadata Explorer* können Sie die Primärschlüssel (Name und Spalteninformationen) einer bestimmten Datenbank-Tabelle abrufen.

Input

Einem Input-XML-Dokument fügen Sie dazu innerhalb eines beliebigen Wurzel-Elements das Element `<primarykeys resulttag="` *`Ergebniselementname`* `"/>` hinzu, in dessen Attribut `resulttag` Sie einen gültigen XML-Namen angeben. Die Angabe einer Tabelle in Element `<table>` ist dabei erforderlich; die Elemente `<catalog>` und `<schema>` sind jedoch optional.

```
<Root>
  <primarykeys resulttag="Ergebniselementname">
     <catalog>Katalog</catalog>
     <schema>DB-Schema</schema>
     <table>DB-Tabelle (Pflichtfeld, darf nicht leer sein)</table>
  </primarykeys>
</Root>
```

Ausgabe

Die XML-Struktur, die vom Adapter ausgegeben wird, besitzt innerhalb des Wurzel-Elements `<Data>` ein XML-Element, dessen Elementname in der Input-Struktur im Attribut `resulttag` angegeben wurde und das die Metadaten zu jedem Primärschlüssel in der Tabelle enthält. Wenn es zu einem Abruf mehrere Ergebnisse gibt (mehrere Primärschlüsselspalten), so werden diese jeweils in einem separaten Element ausgegeben.

```
<Data>
   <Ergebniselementname>
      <Primärschlüssel-Eigenschaft>Wert</Primärschlüssel-Eigenschaft>
      ...
   </Ergebniselementname>
</Data>
```

Erläuterungen (englischsprachig) zu den einzelnen Spalten-Eigenschaften finden Sie hier.

Beispiel

In einem Input-XML-Dokument wird folgende XML-Struktur definiert, um die Primärschlüssel der Tabelle `HISTORY` auszugeben:

```
<Root>
   <primarykeys resulttag="historykeys">
      <schema></schema>
      <table>HISTORY</table>
   </primarykeys>
</Root>
```

Der *JDBC Metadata Explorer* gibt daraufhin z. B. folgendes Ergebnis-XML-Dokument aus:

```
<Data>
   <historykeys>
      <TABLE_CAT />
      <TABLE_SCHEM>APP</TABLE_SCHEM>
      <TABLE_NAME>HISTORY</TABLE_NAME>
      <COLUMN_NAME>PID</COLUMN_NAME>
      <KEY_SEQ>1</KEY_SEQ>
      <PK_NAME>SQL090505161403130</PK_NAME>
   </historykeys>
</Data>
```

10.6.3.5 Fremdschlüssel einer Tabelle abrufen

Mit dem *JDBC Metadata Explorer* können Sie die Fremdschlüssel (Name und Spalteninformationen) einer bestimmten Tabelle abrufen.

Input

Einem Input-XML-Dokument fügen Sie dazu innerhalb eines beliebigen Wurzel-Elements das Element `<foreignkeys resulttag="` *`Ergebniselementname`* `"/>` hinzu, in dessen Attribut `resulttag` Sie einen gültigen XML-Namen angeben. Die Angabe einer Tabelle in Element `<table>` ist dabei erforderlich; die Elemente `<catalog>` und `<schema>` sind jedoch optional.

```
<Root>
  <foreignkeys resulttag="Ergebniselementname">
     <catalog>Katalog</catalog>
     <schema>DB-Schema</schema>
     <table>DB-Tabelle (Pflichtfeld, darf nicht leer sein)</table>
   </foreignkeys>
</Root>
```

Ausgabe

Die XML-Struktur, die vom Adapter ausgegeben wird, enthält innerhalb des Wurzel-Elements `<Data>` ein Element, dessen Name in der Input-Struktur im Attribut `resulttag` angegeben wurde und das die Metadaten zu jedem Fremdschlüssel in der Tabelle enthält. Wenn es zu einem Abruf mehrere Ergebnisse gibt (d. h. mehrere Fremdschlüsselspalten), so werden diese jeweils in einem separaten Element ausgegeben.

```
<Data>
   <Ergebniselementname>
      <Fremdschlüssel-Eigenschaft>Wert</Fremdschlüssel-Eigenschaft>
      ...
   </Ergebniselementname>
</Data>
```

Erläuterungen (englischsprachig) zu den einzelnen Spalten-Eigenschaften finden Sie hier.

Beispiel

In einem Input-XML-Dokument wird folgende XML-Struktur definiert, um die Fremdschlüssel der Tabelle Provider auszugeben:

```
<Root>
   <foreignkeys resulttag="historykeys">
      <table>Provider</table>
   </foreignkeys>
</Root>
```

Der *JDBC Metadata Explorer* gibt daraufhin z. B. folgendes Ergebnis-XML-Dokument aus:

```
<Data>
   <historykeys>
       <PKTABLE_CAT>port</PKTABLE_CAT>
       <PKTABLE_SCHEM>dbo</PKTABLE_SCHEM>
       <PKTABLE_NAME>NETWORK</PKTABLE_NAME>
       <PKCOLUMN_NAME>NETWORK</PKCOLUMN_NAME>
       <FKTABLE_CAT>port</FKTABLE_CAT>
       <FKTABLE_SCHEM>dbo</FKTABLE_SCHEM>
       <FKTABLE_NAME>PROVIDER</FKTABLE_NAME>
       <FKCOLUMN_NAME>NETWORK</FKCOLUMN_NAME>
       <KEY_SEQ>1</KEY_SEQ>
       <UPDATE_RULE>1</UPDATE_RULE>
       <DELETE_RULE>1</DELETE_RULE>
       <FK_NAME>FK_PROVIDER_NETWORK1</FK_NAME>
       <PK_NAME>PK_NETWORK</PK_NAME>
       <DEFERRABILITY>7</DEFERRABILITY>
   </historykeys>
</Data>
```

10.6.3.6 Fremdschlüssel-Referenzen abrufen

Mit dem *JDBC Metadata Explorer* können Sie die Beschreibung der Fremdschlüssel abrufen, die sich in anderen Tabellen befinden und die auf den Primärschlüssel der ausgewählten Tabelle verweisen.

Input

Einem Input-XML-Dokument fügen Sie dazu innerhalb eines beliebigen Wurzel-Elements das Element <exportedkeys resulttag=" *Ergebniselementname* "/> hinzu, in dessen Attribut resulttag Sie einen gültigen XML-Namen angeben. Die Angabe einer Tabelle in Element <table> ist dabei erforderlich; die Elemente <catalog> und <schema> sind jedoch optional.

```
<Root>
  <exportedkeys resulttag="Ergebniselementname">
     <catalog>Katalog</catalog>
     <schema>DB-Schema</schema>
     <table>DB-Tabelle (Pflichtfeld, darf nicht leer sein)</table>
   </exportedkeys>
</Root>
```

Ausgabe

Die XML-Struktur, die vom Adapter ausgegeben wird, enthält innerhalb des Wurzel-Elements `<Data>` ein Element, dessen Name in der Input-Struktur im Attribut `resulttag` angegeben wurde und das die Metadaten zu jedem Fremdschlüssel in anderen Tabellen enthält, der auf den oder die Primärschlüssel der ausgewählten Tabelle verweist. Wenn es mehrere Ergebnisse gibt (d. h. mehrere Fremdschlüssel), so werden diese jeweils in einem separaten Element ausgegeben.

```
<Data>
   <Ergebniselementname>
      <Fremdschlüssel-Eigenschaft>Wert</Fremdschlüssel-Eigenschaft>
      ...
   </Ergebniselementname>
</Data>
```

ⓘ Erläuterungen (englischsprachig) zu den einzelnen Spalten-Eigenschaften finden Sie hier.

Beispiel

In einem Input-XML-Dokument wird folgende XML-Struktur definiert, um die Fremdschlüssel-Referenzen zum Primärschlüssel der Tabelle `Provider` auszugeben:

```
<Root>
   <exportedkeys resulttag="keys">
      <table>Operator</table>
   </exportedkeys>
</Root>
```

Der *JDBC Metadata Explorer* gibt daraufhin z. B. folgendes Ergebnis-XML-Dokument aus:

```
<Data>
   <keys>
      <PKTABLE_CAT>port</PKTABLE_CAT>
      <PKTABLE_SCHEM>dbo</PKTABLE_SCHEM>
      <PKTABLE_NAME>OPERATOR</PKTABLE_NAME>
      <PKCOLUMN_NAME>OPERATOR_ID</PKCOLUMN_NAME>
      <FKTABLE_CAT>port</FKTABLE_CAT>
      <FKTABLE_SCHEM>dbo</FKTABLE_SCHEM>
      <FKTABLE_NAME>ROLEMAP</FKTABLE_NAME>
      <FKCOLUMN_NAME>OPERATOR_ID</FKCOLUMN_NAME>
      <KEY_SEQ>1</KEY_SEQ>
      <UPDATE_RULE>1</UPDATE_RULE>
      <DELETE_RULE>1</DELETE_RULE>
      <FK_NAME>FK_ROLEMAP_OPERATOR</FK_NAME>
      <PK_NAME>PK_OPERATOR</PK_NAME>
      <DEFERRABILITY>7</DEFERRABILITY>
   </keys>
</Data>
```

10.7 EDI

- EDIFACT Adapter
- Unterstützte EDIFACT-Nachrichten-Typen

10.7.1 EDIFACT Adapter

? Unknown Attachment `EDIFACT Adapter`: Liest eine EDIFACT-Nachricht (Rohdaten und Stage 1-3, *DIN ISO 9735*-konform) und gibt sie als UTF-8-kodiertes XML-Dokument aus. Außerdem kann der Adapter eine EDIFACT-Nachricht ausgeben.

Ehemalige Bezeichnung bis Version 4.5.1: *EDIFACT Message Adapter*

Ehemalige Bezeichnung bis Version 5.5.4: *EDIFACT Converter*

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`ReadRaw`*: EDIFACT-Nachricht 1:1 als XML ausgeben • *`ReadStage1`*: EDIFACT-Nachricht als XML mit konkreten Elementnamen einlesen (*Stage 1*) • *`ReadStage2`*: EDIFACT-Nachricht als XML einlesen, in dem die Segmente anhand der Nachrichten organisiert sind (*Stage 2*) • *`ReadStage3`*: EDIFACT-Nachricht als XML einlesen, in dem die Segmente mit der korrekten Hierarchie inklusive verschachtelten Segmentgruppen im Baum gegliedert sind, sofern der Adapter den Nachrichtentyp unterstützt (*Stage 3*) • *`Write`*: EDIFACT-Nachricht schreiben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.edifact.EdifactAdapter`*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis (Input ist *null*).
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Eine EDIFACT-Nachricht vom Typ INVOIC (Rechnung) könnte wie folgt aufgebaut sein:

```
UNA:+,? '
UNB+UNOC:2+FHPEDAL+HUBERGMBH+990802:1557+9908021557'
UNH+INVOIC0001+INVOIC:D:93A:UN'
BGM+380+9908001+9'
DTM+3:19990802:102'
RFF+ON:O0010001'
DTM+4:19999715:102'
NAD+SE++Fahrradhandel Pedal++Wagingerstr. 5+München++81549'
NAD+BY++Huber GmbH++Obstgasse 2+München++81549'
LIN+1++4711.001'
IMD+F++:::Fahrrad, Damen'
QTY+47:1:PCE'
MOA+66:750'
PRI+AAA:750'
LIN+2++4711.002'
IMD+F++:::Luftpumpe, Stand-'
QTY+47:1:PCE'
MOA+66:19,9'
PRI+AAA:19,9'
LIN+3++4711.003'
IMD+F++:::Ersatzventil'
QTY+47:3:PCE'
MOA+66:7,5'
PRI+AAA:2,5'
UNS+S'
MOA+79:777,4'
MOA+124:124,38'
MOA+128:901,78'
TAX+7+VAT+++:::16+S'
UNT+28+INVOIC0001'
UNZ+1+9908021557'
```

Ausgabe

Mit Operation `ReadRaw` werden EDIFACT-Nachrichten 1:1 als XML ausgegeben, z. B.:

```
<EDIFACT decimalSeparator="," segmentTerminator="'"
   releaseIndicator="?" dataSeparator="+" componentSeparator=":">
   <Segment>
      <Component>
         <Data>UNB</Data>
      </Component>
      <Component>
         <Data>UNOC</Data>
         <Data>2</Data>
      </Component>
      <Component>
         <Data>FHPEDAL</Data>
      </Component>
      <Component>
         <Data>HUBERGMBH</Data>
      </Component>
      <Component>
         <Data>990802</Data>
         <Data>1557</Data>
      </Component>
      <Component>
         <Data>9908021557</Data>
      </Component>
   </Segment>
   <!--weitere Segmente-->
</EDIFACT>
```

Mit Operation `ReadStage1` werden EDIFACT-Nachrichten als XML-Dokumente ausgegeben, die bereits statt generischen Elementnamen konkrete Namen verwenden, z. B.:

```
<EDIFACT decimalSeparator="," segmentTerminator="'"
   releaseIndicator="?" dataSeparator="+" componentSeparator=":">
   <UNB>
      <Component>
         <Data>UNOC</Data>
         <Data>2</Data>
      </Component>
      <Component>
         <Data>FHPEDAL</Data>
      </Component>
      <Component>
         <Data>HUBERGMBH</Data>
      </Component>
      <Component>
         <Data>990802</Data>
         <Data>1557</Data>
      </Component>
      <Component>
         <Data>9908021557</Data>
      </Component>
   </UNB>
   <!--weitere Segmente-->
</EDIFACT>
```

Mit Operation `ReadStage2` werden EDIFACT-Nachrichten als XML-Dokumente ausgegeben, in denen die Segmente anhand der Nachrichten organisiert sind, z. B.:

```
<EDIFACT decimalSeparator="," segmentTerminator="'"
   releaseIndicator="?" dataSeparator="+" componentSeparator=":">
   <UNB>
      <Component>
         <Data>UNOC</Data>
         <Data>2</Data>
      </Component>
      <!--weitere Komponenten-->
      <UNH>
         <Component>
            <Data>INVOIC0001</Data>
         </Component>
         (weitere Komponenten)
         <BGM>
            <Component>
               <Data>380</Data>
            </Component>
            <Component>
               <Data>9908001</Data>
            </Component>
            <Component>
               <Data>9</Data>
            </Component>
         </BGM>
         <!--weitere Segmente-->
         <UNT>
            <Component>
               <Data>28</Data>
            </Component>
            <Component>
               <Data>INVOIC0001</Data>
            </Component>
         </UNT>
      </UNH>
      <UNZ>
         <Component>
            <Data>1</Data>
         </Component>
         <Component>
            <Data>99080215557</Data>
         </Component>
      </UNZ>
   </UNB>
</EDIFACT>
```

Mit Operation `ReadStage3` werden EDIFACT-Nachrichten als XML-Dokumente ausgegeben, in denen die Segmente mit der korrekten Hierarchie inklusive verschachtelten Segmentgruppen im Baum gegliedert sind, sofern der Adapter den Nachrichtentyp unterstützt, z. B.:

```
<EDIFACT decimalSeparator="," segmentTerminator="'"
   releaseIndicator="?" dataSeparator="+" componentSeparator=":">
   <UNB>
      <Component>
         <Data>UNOC</Data>
         <Data>2</Data>
      </Component>
      <!--weitere Komponenten-->
      <UNH>
         <Component>
            <Data>INVOIC0001</Data>
         </Component>
         <Component>
            <Data>INVOIC</Data>
            <Data>D</Data>
            <Data>93A</Data>
            <Data>UN</Data>
         </Component>
         <BGM>
            <Component>
               <Data>380</Data>
            </Component>
            <Component>
               <Data>9908001</Data>
            </Component>
            <Component>
               <Data>9</Data>
            </Component>
         </BGM>
         <DTM>
            <Component>
               <Data>3</Data>
               <Data>19990802</Data>
               <Data>102</Data>
            </Component>
         </DTM>
         <!--weitere Segmente-->
         <SegmentGroup22>
            <LIN>
               <Component>
                  <Data>1</Data>
               </Component>
               <Component>
                  <Data />
               </Component>
               <Component>
                  <Data>4711.001</Data>
               </Component>
            </LIN>
            <IMD>
               <Component>
                  <Data>F</Data>
               </Component>
               <Component>
```

```
                  <Data />
               </Component>
               <Component>
                  <Data />
                  <Data />
                  <Data />
                  <Data>Fahrrad, Damen</Data>
               </Component>
            </IMD>
            <QTY>
               <Component>
                  <Data>47</Data>
                  <Data>1</Data>
                  <Data>PCE</Data>
               </Component>
            </QTY>
            <SegmentGroup23>
               <MOA>
                  <Component>
                     <Data>66</Data>
                     <Data>750</Data>
                  </Component>
               </MOA>
            </SegmentGroup23>
            <SegmentGroup25>
               <PRI>
                  <Component>
                     <Data>AAA</Data>
                     <Data>750</Data>
                  </Component>
               </PRI>
            </SegmentGroup25>
         </SegmentGroup22>
         <!--weitere Segmentgruppen-->
         <UNT>
            <Component>
               <Data>28</Data>
            </Component>
            <Component>
               <Data>INVOIC0001</Data>
            </Component>
         </UNT>
      </UNH>
      <UNZ>
         <Component>
            <Data>1</Data>
         </Component>
         <Component>
            <Data>9908021557</Data>
         </Component>
      </UNZ>
   </UNB>
</EDIFACT>
```

Siehe auch:

- Function Adapter
- Unterstützte EDIFACT-Nachrichten-Typen

10.7.2 Unterstützte EDIFACT-Nachrichten-Typen

In der X4 BPMS können mithilfe des EDIFACT Adapter-Adapters eine Reihe von EDIFACT-Nachrichten bearbeitet werden. Die unterstützten Nachrichtentypen werden stetig erweitert.
Derzeit werden folgende Nachrichtentypen und Revisionen unterstützt:

APERAK	• APERAK Rev 1: APERAK*D*95A*UN - APERAK*D*96A*UN • APERAK Rev 2: APERAK*D*96B*UN - APERAK*D*97A*UN • APERAK Rev 3: APERAK*D*97B*UN - APERAK*D*98B*UN • APERAK Rev 4: APERAK*D*99A*UN - APERAK*D*13B*UN
UTILMD	• UTILMD Rev 1: UTILMD*D*01A*UN - UTILMD*D*01B*UN • UTILMD Rev 2: UTILMD*D*01C*UN - UTILMD*D*02A*UN • UTILMD Rev 3: UTILMD*D*02B*UN - UTILMD*D*03B*UN • UTILMD Rev 4: UTILMD*D*04A*UN - UTILMD*D*05B*UN • UTILMD Rev 5: UTILMD*D*06A*UN - UTILMD*D*07A*UN • UTILMD Rev 6: UTILMD*D*07B*UN - UTILMD*D*13B*UN • UTILMD custom: UTILMD*D*02B*ZZ
IFCSUM	• IFCSUM Rev ?? (jedoch ohne IFCSUM*D*93A*UN) • IFCSUM Rev 3 (jedoch ohne IFCSUM*D*95A*UN - IFCSUM*D*95B*UN) • IFCSUM Rev 4 (jedoch ohne IFCSUM*D*96A*UN) • IFCSUM Rev 5: IFCSUM*D*96B*UN • IFCSUM Rev 6: IFCSUM*D*97A*UN • IFCSUM Rev 7: IFCSUM*D*97B*UN • IFCSUM Rev 8: IFCSUM*D*98A*UN • IFCSUM Rev 9: IFCSUM*D*98B*UN - IFCSUM*D*99A*UN • IFCSUM Rev 10: IFCSUM*D*99B*UN • IFCSUM Rev 11: IFCSUM*D*00A*UN • IFCSUM Rev 12: IFCSUM*D*00B*UN - IFCSUM*D*01B*UN • IFCSUM Rev 13: IFCSUM*D*01C*UN - IFCSUM*D*02B*UN • IFCSUM Rev 14: IFCSUM*D*03A*UN - IFCSUM*D*08A*UN • IFCSUM Rev 15: IFCSUM*D*08B*UN - IFCSUM*D*10A*UN • IFCSUM Rev 16: IFCSUM*D*10B*UN - IFCSUM*D*13B*UN

IFTMBF	• IFTMBF Rev ??: IFTMBF*D*93A*UN • IFTMBF Rev 1 (jedoch ohne IFTMBF*D*94A*UN) • IFTMBF Rev 2 (jedoch ohne IFTMBF*D*94B*UN) • IFTMBF Rev 3: IFTMBF*D*95A*UN • IFTMBF Rev 4 (jedoch ohne IFTMBF*D*95B*UN) • IFTMBF Rev 5: IFTMBF*D*96A*UN - IFTMBF*D*96B*UN • IFTMBF Rev 6: IFTMBF*D*97A*UN • IFTMBF Rev 7: IFTMBF*D*97B*UN • IFTMBF Rev 8: IFTMBF*D*98A*UN - IFTMBF*D*99A*UN • IFTMBF Rev 9: IFTMBF*D*99B*UN • IFTMBF Rev 10: IFTMBF*D*00A*UN - IFTMBF*D*08A*UN • IFTMBF Rev 11: IFTMBF*D*08B*UN - IFTMBF*D*10A*UN • IFTMBF Rev 12: IFTMBF*D*10B*UN - IFTMBF*D*13B*UN
IFTMCS	• IFTMCS Rev ??: IFTMCS*D*93A*UN • IFTMCS Rev 1 (jedoch ohne IFTMCS*D*94A*UN) • IFTMCS Rev 2 (jedoch ohne IFTMCS*D*94B*UN) • IFTMCS Rev 3: IFTMCS*D*95A*UN • FTMCS Rev 4 (jedoch ohne IFTMCS*D*95B*UN) • IFTMCS Rev 5: IFTMCS*D*96A*UN - IFTMCS*D*96B*UN • IFTMCS Rev 6: IFTMCS*D*97A*UN • IFTMCS Rev 7: IFTMCS*D*97B*UN • IFTMCS Rev 8: IFTMCS*D*98A*UN - IFTMCS*D*99A*UN • IFTMCS Rev 9: IFTMCS*D*99B*UN • IFTMCS Rev 10: IFTMCS*D*00A*UN - IFTMCS*D*08A*UN • IFTMCS Rev 11: IFTMCS*D*08B*UN - IFTMCS*D*10A*UN • IFTMCS Rev 12: IFTMCS*D*10B*UN - IFTMCS*D*13B*UN

IFTSTA	• IFTSTA Rev ??: IFTSTA*D*93A*UN • IFTSTA Rev 1: IFTSTA*D*94A*UN • IFTSTA Rev 2 (jedoch ohne IFTSTA*D*94B*UN) • IFTSTA Rev 3: IFTSTA*D*95A*UN • IFTSTA Rev 4: IFTSTA*D*95B*UN • IFTSTA Rev 5: IFTSTA*D*96A*UN • IFTSTA Rev 6: IFTSTA*D*96B*UN - IFTSTA*D*97A*UN • IFTSTA Rev 7: IFTSTA*D*97B*UN • IFTSTA Rev 8: IFTSTA*D*98A*UN - IFTSTA*D*99A*UN • IFTSTA Rev 9: IFTSTA*D*99B*UN • IFTSTA Rev 10: IFTSTA*D*00A*UN - IFTSTA*D*00B*UN • IFTSTA Rev 11: IFTSTA*D*01A*UN - IFTSTA*D*01B*UN • IFTSTA Rev 12: IFTSTA*D*01C*UN - IFTSTA*D*02B*UN • IFTSTA Rev 13: IFTSTA*D*03A*UN • IFTSTA Rev 14: IFTSTA*D*03B*UN - IFTSTA*D*08A*UN • IFTSTA Rev 15: IFTSTA*D*08B*UN - IFTSTA*D*12A*UN • IFTSTA Rev 16: IFTSTA*D*12B*UN • IFTSTA Rev 17: IFTSTA*D*13A*UN - IFTSTA*D*13B*UN
INVOIC	• INVOIC Rev 1: INVOIC*D*93A*UN - INVOIC*D*94A*UN • INVOIC Rev 2: INVOIC*D*94B*UN • INVOIC Rev 3: INVOIC*D*95A*UN - INVOIC*D*95B*UN • INVOIC Rev 4: INVOIC*D*96A*UN • INVOIC Rev 5: INVOIC*D*96B*UN • INVOIC Rev 6: INVOIC*D*97A*UN • INVOIC Rev 7: INVOIC*D*97B*UN • INVOIC Rev 8: INVOIC*D*98A*UN • INVOIC Rev 9: INVOIC*D*98B*UN • INVOIC Rev 10: INVOIC*D*99A*UN • INVOIC Rev 11: INVOIC*D*99B*UN - INVOIC*D*00A*UN • INVOIC Rev 12: INVOIC*D*00B*UN • INVOIC Rev 13: INVOIC*D*01A*UN - INVOIC*D*02A*UN • INVOIC Rev 14: INVOIC*D*02B*UN - INVOIC*D*07A*UN • INVOIC Rev 15: INVOIC*D*07B*UN - INVOIC*D*12B*UN • INVOIC Rev 16: INVOIC*D*13A*UN - INVOIC*D*13B*UN
MSCONS	• MSCONS Rev 1: MSCONS*D*97A*UN - MSCONS*D*99A*UN • MSCONS Rev 2: MSCONS*D*99B*UN - MSCONS*D*13B*UN

ORDERS	• ORDERS Rev ??: ORDERS*D*93A*UN • ORDERS Rev 1: ORDERS*D*94A*UN • ORDERS Rev 2: ORDERS*D*94B*UN • ORDERS Rev 3: ORDERS*D*95A*UN • ORDERS Rev 4: ORDERS*D*95B*UN • ORDERS Rev 5: ORDERS*D*96A*UN • ORDERS Rev 6: ORDERS*D*96B*UN • ORDERS Rev 7: ORDERS*D*97A*UN • ORDERS Rev 8: ORDERS*D*97B*UN - ORDERS*D*98A*UN • ORDERS Rev 9: ORDERS*D*98B*UN • ORDERS Rev 10: ORDERS*D*99A*UN • ORDERS Rev 11: ORDERS*D*99B*UN - ORDERS*D*00A*UN • ORDERS Rev 12: ORDERS*D*00B*UN - ORDERS*D*01C*UN • ORDERS Rev 13: ORDERS*D*02A*UN • ORDERS Rev 14: ORDERS*D*02B*UN - ORDERS*D*05B*UN • ORDERS Rev 15: ORDERS*D*06A*UN - ORDERS*D*06B*UN • ORDERS Rev 16: ORDERS*D*07A*UN - ORDERS*D*13B*UN
NOMINT	NOMINT*4*0*EG

ORDRSP	• ORDRSP Rev ??: ORDRSP*D*93A*UN • ORDRSP Rev 1: ORDRSP*D*94A*UN • ORDRSP Rev 2: ORDRSP*D*94B*UN • ORDRSP Rev 3: ORDRSP*D*95A*UN - ORDRSP*D*95B*UN • ORDRSP Rev 4: ORDRSP*D*96A*UN • ORDRSP Rev 5: ORDRSP*D*96B*UN • ORDRSP Rev 6: ORDRSP*D*97A*UN • ORDRSP Rev 7: ORDRSP*D*97B*UN • ORDRSP Rev 8: ORDRSP*D*98A*UN • ORDRSP Rev 9: ORDRSP*D*98B*UN • ORDRSP Rev 10: ORDRSP*D*99A*UN • ORDRSP Rev 11: ORDRSP*D*99B*UN - ORDRSP*D*00A*UN • ORDRSP Rev 12: ORDRSP*D*00B*UN - ORDRSP*D*01C*UN • ORDRSP Rev 13: ORDRSP*D*02A*UN • ORDRSP Rev 14: ORDRSP*D*02B*UN - ORDRSP*D*05B*UN • RDRSP Rev 15: ORDRSP*D*06A*UN • ORDRSP Rev 16: ORDRSP*D*06B*UN - ORDRSP*D*07B*UN • ORDRSP Rev 17: ORDRSP*D*08A*UN - ORDRSP*D*13B*UN
DESADV	• DESADV Rev ??: DESADV*D*93A*UN • DESADV Rev 1: DESADV*D*94A*UN • DESADV Rev 2: DESADV*D*94B*UN • DESADV Rev 3: DESADV*D*95A*UN • DESADV Rev 4: DESADV*D*95B*UN • DESADV Rev 5: DESADV*D*96A*UN • DESADV Rev 6: DESADV*D*96B*UN - DESADV*D*97B*UN • DESADV Rev 7: DESADV*D*98A*UN • DESADV Rev 8: DESADV*D*98B*UN • DESADV Rev 9: DESADV*D*99A*UN • DESADV Rev 10: DESADV*D*99B*UN - DESADV*D*01C*UN • DESADV Rev 11: DESADV*D*02A*UN • DESADV Rev 12: DESADV*D*02B*UN - DESADV*D*10B*UN • DESADV Rev 13: DESADV*D*11A*UN • DESADV Rev 14: DESADV*D*11B*UN - DESADV*D*13B*UN

10.8 E-Mail

- IMAP
- MIME Multipart
- MIME Multipart Reader
- POP/SMTP

10.8.1 IMAP

`IMAP`: Empfängt, kopiert/verschiebt oder löscht eine E-Mail-Nachricht in einem IMAP-Postfach (Internet Message Access Protocol), setzt ein Flag oder listet Nachrichten-Header in einem Ordner eines IMAP-Postfachs.

Der Adapter unterstützt mit S/MIME verschlüsselte/signierte Nachrichten sowie die SSL-Verschlüsselung der Verbindung zum IMAP-Server (*IMAPS*). Aus einer Properties-Datei können beliebige *Java Mail Properties* verarbeitet werden.

Ehemalige Bezeichnung bis Version 4.5.1: *IMAP Connection Adapter*

Ehemalige Bezeichnung bis Version 5.5.4: *IMAP Connector*

Die Standard-Blockgröße (16 KB) bei der Nachrichtenübertragung kann über die JavaMail-System-Eigenschaft `mail.imap.fetchsize` in der Laufzeitumgebung des X4 Servers (i.d.R. Java-VM mit Applikations-Server) gesetzt werden, um das Herunterladen der E-Mails via IMAP optimiert.
Beispiel für das Startskript von *JBoss*, die Blockgröße wird auf 5 MB gesetzt:

```
@set JAVA_OPTS=%JAVA_OPTS% -Dmail.imap.fetchsize=5242880
```

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`Receive Message`*: Nachricht mit der in Parameter `messageID` angegeben ID als UTF-8-kodiertes XML-Dokument empfangen • *`Receive EML Message`*: Nachricht mit der in Parameter `messageID` angegeben ID empfangen und als `.eml`-Datei (MS Outlook) speichern. ⓘ Eine Weiterleitung der empfangenen Nachricht ist mit direkter Anbindung des IMAP-Adapters an den POP/SMTP-Adapter möglich. • *`List Messages`*: Nachrichten-Header eines IMAP-Ordners, der in Parameter `folder` angegeben wird, in einem UTF-8-kodierten XML-Dokument auflisten • *`Copy Message`*: Nachricht, deren Nachrichten-ID in Parameter `messageID` angegeben wird, in den IMAP-Ordner kopieren, der in Parameter `folder` angegeben wird • *`Move Message`*: Nachricht, deren Nachrichten-ID in Parameter `messageID` angegeben wird, in den IMAP-Ordner verschieben, der in Parameter `folder` angegeben wird • *`Delete Message`*: Nachricht, deren Nachrichten-ID in Parameter `messageID` angegeben wird, vom IMAP-Server löschen • *`Set Flag`*: Nachricht, deren Nachrichten-ID in Parameter `messageID` angegeben wird, mit einem oder mehreren Flags kennzeichnen, die in Parameter `flags` angegeben werden

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.imap.Adapter*: Hauptklasse (Standard)
`imapServer`	IP-Adresse oder Host-Name des IMAP-Servers **Mögliche Werte:** Beliebige gültige Zeichenkette (z. B. *imap.1und1.de*)
`login`	Benutzername für das verwendete E-Mail-Konto **Mögliche Werte:** Beliebige Zeichenkette
`password`	Passwort für das verwendete E-Mail-Konto **Mögliche Werte:** Beliebige Zeichenkette
`oAuth2`	Legt fest, ob das OAuth 2.0-Protokoll zum Log-In verwendet werden soll. **Mögliche Werte:** • *ja*: oAuth2 wird zum Log-in verwendet • *nein*: oAuth2 wird nicht zum Log-in verwendet ⚠ Wenn OAuth 2.0 verwendet werden soll, • muss die SSL-Verschlüsselung für die Verbindung zum IMAP-Server durch setzen des Parameters `ssl` aktiviert sein. • muss beim Parameter `password` der Access Token eingetragen werden.

`folder`	Zielordner auf dem IMAP-Server (wenn die Eigenschaft Operation auf *Copy Message* oder *Move Message* gesetzt ist) oder Ordner, dessen Nachrichten aufgelistet werden soll (wenn die Eigenschaft Operation auf *List Messages* gesetzt ist) **Mögliche Werte:** Vorhandener Ordnername (Groß-/Kleinschreibung beachten) Häufig verwendet: • INBOX • Drafts • Sent Messages • Junk • Deleted Messages • Archive • Notes
`srcFolder`	Quellordner für die Aktionen *Copy Message*, *Move Message*, *Receive Message*, *Set Flag* und *Delete Message*. **Mögliche Werte:** Vorhandener Ordnername (Groß-/Kleinschreibung beachten) Häufig verwendet: • INBOX • Drafts • Sent Messages • Junk • Deleted Messages • Archive • Notes
`flags`	IMAP-Flag(s) zur Kennzeichnung einer E-Mail-Nachricht; Mehrere Flags werden durch Semikolon (*;*) getrennt (wenn die Eigenschaft Operation auf *Set Flag* gesetzt ist) **Mögliche Werte:** Beliebiges Flag (z. B. *Answered*)
`messageID`	Nachrichten-ID der E-Mail inklusive Ordnerangabe **Mögliche Werte:** Ordnername und Nachrichten-ID (mit spitzen Klammern) (z. B. *INBOX/<831921101.31255094008434.JavaMail.javamailuser@localhost>*)
`uid`	Nachrichten-ID auf dem IMAP-Server **Mögliche Werte:** Beliebige ganze Zahl

`maxMessages`	Anzahl der aufzulistenden Nachrichten begrenzen, wenn die Eigenschaft Operation auf *List Messages* gesetzt ist. Die Auflistung beginnt mit den ältesten Nachrichten im Ordner. **Mögliche Werte:** Beliebige ganze Zahl
`base64`	Anhänge von Nachrichten (Multipart-Teile) immer Base64-kodieren (wenn die Eigenschaft Operation auf *Receive Message* gesetzt ist) **Mögliche Werte:** • *ja*: Alle Multipart-Teile Base64-kodieren • *nein*: Nicht Base64-kodieren (Standard)
`encrypt`	Nachrichten sind mit S/MIME verschlüsselt (*Multipart/Encrypted*) **Mögliche Werte:** • *ja*: S/MIME-Verschlüsselung aktivieren (erfordert die Angabe zur Schlüssel-/Zertifikatsdatei in Parameter Pkcs12 und der Passphrase in Parameter passphrase) • *nein*: Keine Verschlüsselung (Standard)
`sign`	Nachrichten sind mit S/MIME signiert (*Multipart/Signed*) **Mögliche Werte:** • *ja*: S/MIME-Signatur verarbeiten (erfordert die Angabe zur Schlüssel-/Zertifikatsdatei in Parameter Pkcs12 und der Passphrase in Parameter passphrase) • *nein*: Keine Signatur (Standard)
`Pkcs12`	Pfad zur einer gültigen Schlüssel-/Zertifikatsdatei im *PKCS#12*-Format **Mögliche Werte:** Beliebige gültige Pfadangabe im lokalen Dateisystem des *X4 Servers* (z. B. *%USER_COLLECTION\Project\Folder\Key.p12*)
`passphrase`	Passwort/Passphrase zur Schlüssel-/Zertifikatsdatei im *PKCS#12*-Format, die Sie in Parameter Pkcs12 angegeben haben **Mögliche Werte:** Beliebige gültige Zeichenkette

`port`	Port des IMAP-Servers **Mögliche Werte:** • Beliebige ganze Zahl (z. B. *993*) • *(kein Wert)*: Standard-Port *143* verwenden
`ssl`	SSL-Verschlüsselung für die Verbindung zum IMAP-Server aktivieren **Mögliche Werte:** • *Ja*: IMAP über SSL (IMAPS) verwenden • *Nein*: Verbindung nicht verschlüsseln (Standard)
`propertiesURL`	URL zu einer Properties-Datei (XML oder Text), die beliebige *Java Mail Properties* enthalten kann **Mögliche Werte:** • Beliebige gültige URL zu einer Properties-Datei (z. B. *xstore://Project/Folder/ mail.properties*) • *(leer)*: Keine *Java Mail Properties* verarbeiten
`propertiesType`	Format der Properties-Datei, die in Parameter `propertiesURL` angegeben wurde **Mögliche Werte:** • *`TEXT`*: Als Text-Properties interpretieren (Standard) • *`XML`*: Als XML-Properties interpretieren

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde durchgeführt, es wurde aber kein Ergebnis gefunden oder keine Daten geändert.
`-1`	Die Operation konnte nicht ausgeführt werden.

10.8.2 MIME Multipart

`MIME Multipart`: Ermöglicht das Auslesen und das komfortable Erzeugen von MIME-Multipart-Dokumenten – primär für den Einsatzbereich HTTP-Kommunikation (z. B. ReSTful Webservices). Der Adapter ist zudem für die E-Mail-Kommunikation bzw. für den Umgang mit E-Mail-Anhängen nutzbar.

Der Adapter unterstützt Sie bei der Arbeit mit MIME-Multipart-Dokumenten. Er konvertiert MIME-Dokumente in eine XML-Repräsentation und die XML-Repräsentation in MIME-Dokumente.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`FromBytesWithHeaders:`* aus einem MIME-Dokument ein MIME-XML-Dokument erzeugen • *`FromRestMetaXml`* : aus einem ReST-Input-XML ein entsprechendes MIME-XML-Dokument erzeugen • *`ModifyRestMetaXml`* : aus einem ReST-Input-XML ein modifiziertes ReST-Input-XML erzeugen • *`ToRestMetaXml`* : aus einem MIME-XML-Dokument ein ReST-Output-XML mit HTTP-Statuscode `200` erzeugen • *`ToBytesWithHeaders`* : aus einem MIME-XML-Dokument ein MIME-Dokument erzeugen • *`FromBytesWithBoundaryFromFirstLine`* : aus einem MIME-Body ein MIME-XML-Dokument erzeugen • *`ToBytesWithBoundaryInFirstLine`* : aus einem MIME-XML-Dokument einen MIME-Body ohne Präambel und ohne Epilog erzeugen • *`FlattenXml`* : aus einem MIME-XML-Dokument ein "flaches" MIME-XML-Dokument erzeugen (ggf. wird Multipart-Baum als einzelner Part serialisiert) • *`PartFromX4Document:`* Verwandelt ein beliebiges eingehendes X4-Document in ein Part in der Mime-XML-Struktur

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.mimemultipart.Adapter*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.8.2.1 Operation FromBytesWithHeaders

Mit der Adapter-Operation *`FromBytesWithHeaders`* lassen sich die Bestandteile eines MIME-Dokuments (z.B. eine E-Mail im EML-Format) verarbeiten, z.B. die Anhänge der E-Mail.

Input: Beispiel eines MIME(-Multipart)-Dokuments

```
From: John Doe <example@example.com>
MIME-Version: 1.0
Content-Type: multipart/mixed;
        boundary="XXXXboundary text"

This is a multipart message in MIME format.

--XXXXboundary text
Content-Type: text/plain

this is the body text

--XXXXboundary text
Content-Type: text/plain;
Content-Disposition: attachment;
        filename="test.txt"

this is the attachment text

--XXXXboundary text--
```

Output: Beispiel eines erzeugten MIME(-Multipart)-XML-Dokument

```
<Multipart>
    <Header>
        <Name>from</Name>
        <Value>John Doe &lt;example@example.com&gt;</Value>
    </Header>
    <Header>
        <Name>mime-version</Name>
        <Value>1.0</Value>
    </Header>
    <Header>
        <Name>content-type</Name>
        <Value>multipart/mixed; boundary="XXXXboundary text"</Value>
        <Type>multipart/mixed</Type>
        <Parameter>
            <Name>boundary</Name>
            <Value>XXXXboundary text</Value>
        </Parameter>
    </Header>
    <Prolog>VGhpcyBpcyBhIG11bHRpcGFydCBtZXNzYWdlIGluIE1JTUUgZm9ybWF0Lg0K</Prolog>
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>text/plain</Value>
            <Type>text/plain</Type>
        </Header>
        <Content encoding="base64">dGhpcyBpcyB0aGUgYm9keSB0ZXh0DQo=</Content>
    </Part>
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>text/plain;</Value>
            <Type>text/plain</Type>
        </Header>
        <Header>
            <Name>content-disposition</Name>
            <Value>attachment; filename="test.txt"</Value>
            <Disposition>attachment</Disposition>
            <Parameter>
                <Name>filename</Name>
                <Value>test.txt</Value>
            </Parameter>
        </Header>
        <Content encoding="base64">dGhpcyBpcyB0aGUgYXR0YWNobWVudCB0ZXh0DQo=</Content>
    </Part>
    <Epilog />
</Multipart>
```

Wenn das Input-Dokument lediglich ein MIME-Dokument enthält, wird im Output-XML-Dokument nur der entsprechende Part in einem Wurzel-Element `<Part/>` ausgegeben.

10.8.2.2 Operation FromRestMetaXml

Mit der Adapter-Operation `FromRestMetaXml` lassen sich XML-Dokumente im *MetaXml* -Format verarbeiten und als MIME-XML ausgeben. Dieses Input-XML-Format wird erzeugt, wenn ein X4-Prozess über den *X4 ReST Starter* aufgerufen wurde und dieser den Input-Handler *`HTTP-MetaXml`* verwendet.

Einsatzbeispiele:

- Datei-Upload mit Hilfe eines HTML-Formulars: wenn Daten mit dem Content-Type `multipart/form-data` verarbeitet werden sollen.
- Webservices-Kommunikation, bei der sämtliche (Meta-)Daten des HTTP-Requests für die Verarbeitung relevant sind.

Input: Beispiel eines X4 ReST-Input-Dokuments

```
<Request>
    <Path>
        <Method>POST</Method>
        <FullPath>http://localhost:8080/X4/httpstarter/ReST/httpService/simple</
FullPath>
        <Server>http://localhost:8080/</Server>
        <RequestPath>/X4/httpstarter/ReST/httpService/simple</RequestPath>
        <ContextPrefix>/X4/httpstarter/ReST</ContextPrefix>
        <PathInfo>/httpService/simple</PathInfo>
        <Parts>
            <Constant name="httpService" />
            <Constant name="simple" />
        </Parts>
    </Path>
    <Connection>
        <Scheme>http</Scheme>
        <Remote>
            <Address>0:0:0:0:0:0:0:1</Address>
            <Port>51181</Port>
        </Remote>
        <Local>
            <Address>0:0:0:0:0:0:0:1</Address>
            <Port>8080</Port>
        </Local>
        <Server>
            <Name>localhost</Name>
            <Port>8080</Port>
        </Server>
    </Connection>
    <Headers>
        <Header>
            <Name>accept</Name>
            <Value>text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</
Value>
        </Header>
        <Header>
            <Name>user-agent</Name>
            <Value>Mozilla/5.0 (Windows NT 6.1; WOW64; rv:43.0) Gecko/20100101
Firefox/43.0
            </Value>
        </Header>
        <Header>
            <Name>connection</Name>
            <Value>keep-alive</Value>
        </Header>
        <Header>
            <Name>host</Name>
            <Value>localhost:8080</Value>
        </Header>
        <Header>
            <Name>accept-language</Name>
```

```
            <Value>de,en-US;q=0.7,en;q=0.3</Value>
        </Header>
        <Header>
            <Name>accept-encoding</Name>
            <Value>gzip, deflate</Value>
        </Header>
        <Header>
            <Name>content-length</Name>
            <Value>546</Value>
        </Header>
        <Header>
            <Name>content-type</Name>
            <Value>multipart/form-data;
boundary=--------------------------2109532091963
            </Value>
        </Header>
    </Headers>
    <Query />
    <Security />
    <Body encoding="base64">
        LS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0yMTA5NTMyMDkxOTYzDQpDb250ZW50LURpc3Bv
        c2l0aW9uOiBmb3JtLWRhdGE7IG5hbWU9IkZlbGQxIg0KDQplcnN0ZXMgRmVsZA0KLS0tLS0t
        LS0tLS0tLS0tLS0tLS0tLS0tLS0yMTA5NTMyMDkxOTYzDQpDb250ZW50LURpc3Bvc2l0aW9uOiBm
        b3JtLWRhdGE7IG5hbWU9IkZlbGQyIg0KDQp6d2VpdGVzIEZlbGQNCi0tLS0tLS0tLS0tLS0t
        LS0tLS0tLS0tLS0tMjEwOTUzMjA5MTk2Mw0KQ29udGVudC1EaXNwb3NpdGlvbjogZm9ybS1kYXRh
        OyBuYW1lPSJhcmVhIg0KDQpEaWUgR3Jv32UgRmzkY2hlDQoNCmJsdWINCg0KLS0tLQ0KDQptb3Jl
        IGJsdWINCg0KDQotLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLTIxMDk1MzIwOTE5NjMNCkNv
        bnRlbnQtRGlzcG9zaXRpb246IGZvcm0tZGF0YTsgbmFtZT0iRGF0ZWkiOyBmaWxlbmFtZT0iIg0K
        Q29udGVudC1UeXBlOiBhcHBsaWNhdGlvbi9vY3RldC1zdHJlYW0NCg0KDQotLS0tLS0tLS0tLS0t
        LS0tLS0tLS0tLS0tLS0tLTIxMDk1MzIwOTE5NjMtLQ0K
    </Body>
</Request>
```

Output: Beispiel eines erzeugten MIME(-Multipart)-XML-Dokuments

```
<Multipart>
    <Header>
        <Name>accept</Name>
        <Value>text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</Value>
    </Header>
    <Header>
        <Name>user-agent</Name>
        <Value>Mozilla/5.0 (Windows NT 6.1; WOW64; rv:43.0) Gecko/20100101 Firefox/
43.0</Value>
    </Header>
    <Header>
        <Name>connection</Name>
        <Value>keep-alive</Value>
    </Header>
    <Header>
        <Name>host</Name>
        <Value>localhost:8080</Value>
    </Header>
    <Header>
        <Name>accept-language</Name>
        <Value>de,en-US;q=0.7,en;q=0.3</Value>
    </Header>
    <Header>
        <Name>accept-encoding</Name>
        <Value>gzip, deflate</Value>
    </Header>
    <Header>
        <Name>content-length</Name>
        <Value>546</Value>
    </Header>
    <Header>
        <Name>content-type</Name>
        <Value>multipart/form-data;
boundary=---------------------------2109532091963</Value>
        <Type>multipart/form-data</Type>
        <Parameter>
            <Name>boundary</Name>
            <Value>---------------------------2109532091963</Value>
        </Parameter>
    </Header>
    <Prolog />
    <Part>
        <Header>
            <Name>content-disposition</Name>
            <Value>form-data; name="Feld1"</Value>
            <Disposition>form-data</Disposition>
            <Parameter>
                <Name>name</Name>
                <Value>Feld1</Value>
            </Parameter>
        </Header>
```

```
        <Content encoding="base64">ZXJzdGVzIEZlbGQ=</Content>
    </Part>
    <Part>
        <Header>
            <Name>content-disposition</Name>
            <Value>form-data; name="Feld2"</Value>
            <Disposition>form-data</Disposition>
            <Parameter>
                <Name>name</Name>
                <Value>Feld2</Value>
            </Parameter>
        </Header>
        <Content encoding="base64">endlaXRlcyBGZWxk</Content>
    </Part>
    <Part>
        <Header>
            <Name>content-disposition</Name>
            <Value>form-data; name="area"</Value>
            <Disposition>form-data</Disposition>
            <Parameter>
                <Name>name</Name>
                <Value>area</Value>
            </Parameter>
        </Header>
        <Content encoding="base64">
            RGllIEdyb99lIEZs5GNoZQ0KDQpibHViDQoNCi0tLS0NCg0KbW9yZSBibHViDQoNCg==
        </Content>
    </Part>
    <Part>
        <Header>
            <Name>content-disposition</Name>
            <Value>form-data; name="Datei"; filename=""</Value>
            <Disposition>form-data</Disposition>
            <Parameter>
                <Name>name</Name>
                <Value>Datei</Value>
            </Parameter>
            <Parameter>
                <Name>filename</Name>
                <Value />
            </Parameter>
        </Header>
        <Header>
            <Name>content-type</Name>
            <Value>application/octet-stream</Value>
            <Type>application/octet-stream</Type>
        </Header>
        <Content encoding="base64" />
    </Part>
    <Epilog />
</Multipart>
```

10.8.2.3 Operation ModifyRestMetaXml

Mit der Adapter-Operation `ModifyRestMetaXml` lassen sich analog zu Operation `FromRestMetaXml` die vom *X4 ReST Starter* erzeugte *MetaXml*-Dokumente verarbeiten, jedoch besitzt das ausgegebene Dokument eine veränderte Struktur:

- Der Header-Bereich (`/Request/Headers`) wird entfernt.
- Das Body-Element (`/Request/Body`) wird durch die MIME-XML-Struktur ersetzt.

Input: Rest-Input-XML-Dokument, siehe Input für Operation FromRestMetaXml)

Output: Beispiel eines modifizierten X4 ReST-Input-Dokuments

```
<Request>
    <Path>
        <Method>POST</Method>
        <FullPath>http://localhost:8080/X4/httpstarter/ReST/httpService/simple</
FullPath>
        <Server>http://localhost:8080/</Server>
        <RequestPath>/X4/httpstarter/ReST/httpService/simple</RequestPath>
        <ContextPrefix>/X4/httpstarter/ReST</ContextPrefix>
        <PathInfo>/httpService/simple</PathInfo>
        <Parts>
            <Constant name="httpService" />
            <Constant name="simple" />
        </Parts>
    </Path>
    <Connection>
        <Scheme>http</Scheme>
        <Remote>
            <Address>0:0:0:0:0:0:0:1</Address>
            <Port>51181</Port>
        </Remote>
        <Local>
            <Address>0:0:0:0:0:0:0:1</Address>
            <Port>8080</Port>
        </Local>
        <Server>
            <Name>localhost</Name>
            <Port>8080</Port>
        </Server>
    </Connection>
    <Query />
    <Security />
    <Multipart>
        <Header>
            <Name>accept</Name>
            <Value>text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</
Value>
        </Header>
        <Header>
            <Name>user-agent</Name>
            <Value>Mozilla/5.0 (Windows NT 6.1; WOW64; rv:43.0) Gecko/20100101
Firefox/43.0
            </Value>
        </Header>
        <Header>
            <Name>connection</Name>
            <Value>keep-alive</Value>
        </Header>
        <Header>
            <Name>host</Name>
            <Value>localhost:8080</Value>
        </Header>
```

```
        <Header>
            <Name>accept-language</Name>
            <Value>de,en-US;q=0.7,en;q=0.3</Value>
        </Header>
        <Header>
            <Name>accept-encoding</Name>
            <Value>gzip, deflate</Value>
        </Header>
        <Header>
            <Name>content-length</Name>
            <Value>546</Value>
        </Header>
        <Header>
            <Name>content-type</Name>
            <Value>multipart/form-data;
boundary=---------------------------2109532091963
            </Value>
            <Type>multipart/form-data</Type>
            <Parameter>
                <Name>boundary</Name>
                <Value>---------------------------2109532091963</Value>
            </Parameter>
        </Header>
        <Prolog />
        <Part>
            <Header>
                <Name>content-disposition</Name>
                <Value>form-data; name="Feld1"</Value>
                <Disposition>form-data</Disposition>
                <Parameter>
                    <Name>name</Name>
                    <Value>Feld1</Value>
                </Parameter>
            </Header>
            <Content encoding="base64">ZXJzdGVzIEZlbGQ=</Content>
        </Part>
        <Part>
            <Header>
                <Name>content-disposition</Name>
                <Value>form-data; name="Feld2"</Value>
                <Disposition>form-data</Disposition>
                <Parameter>
                    <Name>name</Name>
                    <Value>Feld2</Value>
                </Parameter>
            </Header>
            <Content encoding="base64">endlaXRlcyBGZWxk</Content>
        </Part>
        <Part>
            <Header>
                <Name>content-disposition</Name>
                <Value>form-data; name="area"</Value>
                <Disposition>form-data</Disposition>
                <Parameter>
                    <Name>name</Name>
```

```
                    <Value>area</Value>
                </Parameter>
            </Header>
            <Content encoding="base64">
                RGllIEdyb99lIEZs5GNoZQ0KDQpibHViDQoNCi0tLS0NCg0KbW9yZSBibHViDQoNCg==
            </Content>
        </Part>
        <Part>
            <Header>
                <Name>content-disposition</Name>
                <Value>form-data; name="Datei"; filename=""</Value>
                <Disposition>form-data</Disposition>
                <Parameter>
                    <Name>name</Name>
                    <Value>Datei</Value>
                </Parameter>
                <Parameter>
                    <Name>filename</Name>
                    <Value />
                </Parameter>
            </Header>
            <Header>
                <Name>content-type</Name>
                <Value>application/octet-stream</Value>
                <Type>application/octet-stream</Type>
            </Header>
            <Content encoding="base64" />
        </Part>
        <Epilog />
    </Multipart>
</Request>
```

10.8.2.4 Operation ToRestMetaXml

Mit der Adapter-Operation `ToRestMetaXml` lässt sich ein MIME-XML-Dokument einlesen und im *MetaXML*-Format erzeugen. Dieses XML-Format wird typischerweise als Ergebnis eines X4-Prozesses ausgegeben, der über den *X4 ReST Starter* aufgerufen wird, wenn dieser den Output-Handler *`HTTP-MetaXml`* verwendet.

Input: Beispiel eines X4 ReST-Input-Dokuments

```
<Multipart>
    <Header>
        <Name>content-transfer-encoding</Name>
        <Value>binary</Value>
    </Header>
    <Header>
        <Name>content-type</Name>
        <Value>multipart/related; type="application/xop+xml";
            start="&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;";
            start-info="text/xml"</Value>
        <Type>multipart/related</Type>
        <Parameter>
            <Name>type</Name>
            <Value>application/xop+xml</Value>
        </Parameter>
        <Parameter>
            <Name>start</Name>
            <Value>&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;</Value>
        </Parameter>
        <Parameter>
            <Name>start-info</Name>
            <Value>text/xml</Value>
        </Parameter>
    </Header>
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>application/xop+xml; charset=UTF-8; type="text/xml"</Value>
            <Type>application/xop+xml</Type>
            <Parameter>
                <Name>charset</Name>
                <Value>UTF-8</Value>
            </Parameter>
            <Parameter>
                <Name>type</Name>
                <Value>text/xml</Value>
            </Parameter>
        </Header>
        <Header>
            <Name>content-transfer-encoding</Name>
            <Value>binary</Value>
        </Header>
        <Header>
            <Name>content-id</Name>
            <Value>&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;</Value>
            <Id asUrl="cid:0@4cd748bf-d5ea-4d2f ...</Id>
        </Header>
        <Content encoding="base64">PD94bWwg ...</Content>
    </Part>
</Multipart>
```

Output: Beispiel eines erzeugten X4 ReST-Output-Dokument

```
<Response>
    <Status code="200"/>
    <Headers>
        <Header>
            <Name>content-transfer-encoding</Name><Value>binary</Value>
        </Header>
        <Header>
            <Name>content-type</Name>
            <Value>multipart/related; type="application/xop+xml";
                    start="&lt;0@4cd748bf-d5ea-4d2f ... &gt;"; start-info="text/
xml";
                boundary=37aa08a1-cbc3-4143-b565-aae4ff9007d8</Value>
        </Header>
    </Headers>
    <Content encoding="base64">LS0zN2FhM ... </Content>
</Response>
```

10.8.2.5 Operation ToBytesWithHeaders

Mit der Adapter-Operation `ToBytesWithHeaders` lässt sich ein MIME-Dokument mit allen benötigten Header-Informationen erzeugen, d.h. sämtliche Header müssen selbst erzeugt werden.

> ⚠ Wenn im Element `Content` das Attribut `encoding="base64"` gesetzt ist, muss der Anhang zweimal Base64 kodiert werden.

Anwendungsfall: Erzeugen einer HTML-E-Mail mit eingebetteten Bildern

Input: Beispiel eines MIME-XML-Dokuments

```
<Multipart>
    <Header>
        <Name>content-transfer-encoding</Name>
        <Value>binary</Value>
    </Header>
    <Header>
        <Name>content-type</Name>
        <Value>multipart/related; type="application/xop+xml";
            start="&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;";
            start-info="text/xml"</Value>
        <Type>multipart/related</Type>
        <Parameter>
            <Name>type</Name>
            <Value>application/xop+xml</Value>
        </Parameter>
        <Parameter>
            <Name>start</Name>
            <Value>&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;</Value>
        </Parameter>
        <Parameter>
            <Name>start-info</Name>
            <Value>text/xml</Value>
        </Parameter>
    </Header>
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>application/xop+xml; charset=UTF-8; type="text/xml"</Value>
            <Type>application/xop+xml</Type>
            <Parameter>
                <Name>charset</Name>
                <Value>UTF-8</Value>
            </Parameter>
            <Parameter>
                <Name>type</Name>
                <Value>text/xml</Value>
            </Parameter>
        </Header>
        <Header>
            <Name>content-transfer-encoding</Name>
            <Value>binary</Value>
        </Header>
        <Header>
            <Name>content-id</Name>
            <Value>&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;</Value>
            <Id asUrl="cid:0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d">
                0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d</Id>
        </Header>
        <Content encoding="base64">PD94bWwgd ... </Content>
    </Part>
</Multipart>
```

Output: Beispiel eines erzeugten MIME-Dokuments

```
content-transfer-encoding: binary
content-type: multipart/related; type="application/xop+xml"; start="<0@4cd748bf-
d5ea-4d2f-89cd-18eb54442e0d>"; start-info="text/xml"; boundary=1e66b8da-5b2c-425e-
b502-7ffdefbe4283

--1e66b8da-5b2c-425e-b502-7ffdefbe4283
content-type: application/xop+xml; charset=UTF-8; type="text/xml"
content-transfer-encoding: binary
content-id: <0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d>


<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:stoc="http://example.com/stockquote.xsd">
    <soapenv:Header/>
    <soapenv:Body>
        <blub>SGFsbG8gSGFsbG8g ...</blub>
        <bla>SGFsbG8=</bla>
        <foo>SGFsbG8=</foo>
        <bar>SGFsbG8=</bar>
    </soapenv:Body>
</soapenv:Envelope>
--1e66b8da-5b2c-425e-b502-7ffdefbe4283--
```

10.8.2.6 Operation FromBytesWithBoundaryFromFirstLine

Mit der Adapter-Operation `FromBytesWithBoundaryFromFirstLine` lassen sich analog zu Adapter-Operation `FromBytesWithHeaders` Bestandteile eines MIME-Dokuments (z.B. eine E-Mail im EML-Format) in einem ausgegebenen XML-Dokument verarbeiten, jedoch wird hier kein Header-Block erwartet und es wird direkt der MIME-Body verarbeitet – vorausgesetzt es handelt sich um eine gültige Multipart-Boundary. Ansonsten wird der Input als Part ohne Header im erzeugten XML-Dokument ausgegeben.

Anwendungsfall: SOAP über JMS. Hier wird eine Byte-Message verwendet, die einen MIME-Body ohne MIME-Header enthält. Um MTOM auch hier verwenden zu können, wird ein MIME-Multipart-Body ohne Präambel verwendet.

Input: Beispiel eines MIME-Bodys

```
--XXXXboundary text
Content-Type: text/plain

this is the body text
--XXXXboundary text
Content-Type: text/plain;
Content-Disposition: attachment;
        filename="test.txt"

this is the attachment text
--XXXXboundary text--
```

Output: Beispiel eines erzeugten MIME-XML-Dokuments

```
<Multipart>
    <Prolog />
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>text/plain</Value>
            <Type>text/plain</Type>
        </Header>
        <Content encoding="base64">dGhpcyBpcyB0aGUgYm9keSB0ZXh0DQo=</Content>
    </Part>
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>text/plain;</Value>
            <Type>text/plain</Type>
        </Header>
        <Header>
            <Name>content-disposition</Name>
            <Value>attachment; filename="test.txt"</Value>
            <Disposition>attachment</Disposition>
            <Parameter>
                <Name>filename</Name>
                <Value>test.txt</Value>
            </Parameter>
        </Header>
        <Content encoding="base64">dGhpcyBpcyB0aGUgYXR0YWNobWVudCB0ZXh0DQo=</Content>
    </Part>
    <Epilog />
</Multipart>
```

10.8.2.7 Operation ToBytesWithBoundaryInFirstLine

Mit der Adapter-Operation `ToBytesWithBoundaryInFirstLine` lässt sich ein MIME-Dokument ohne Header-Block und bei Multiparts ohne Präambel und ohne Epilog erzeugen.

Anwendungsfall: SOAP über JMS (mit MTOM)

Input: Beispiel eines MIME-XML-Dokuments

```
<Multipart>
    <Header>
        <Name>content-transfer-encoding</Name>
        <Value>binary</Value>
    </Header>
    <Header>
        <Name>content-type</Name>
        <Value>multipart/related; type="application/xop+xml";
            start="&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;";
            start-info="text/xml"</Value>
        <Type>multipart/related</Type>
        <Parameter>
            <Name>type</Name>
            <Value>application/xop+xml</Value>
        </Parameter>
        <Parameter>
            <Name>start</Name>
            <Value>&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;</Value>
        </Parameter>
        <Parameter>
            <Name>start-info</Name>
            <Value>text/xml</Value>
        </Parameter>
    </Header>
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>application/xop+xml; charset=UTF-8; type="text/xml"</Value>
            <Type>application/xop+xml</Type>
            <Parameter>
                <Name>charset</Name>
                <Value>UTF-8</Value>
            </Parameter>
            <Parameter>
                <Name>type</Name>
                <Value>text/xml</Value>
            </Parameter>
        </Header>
        <Header>
            <Name>content-transfer-encoding</Name>
            <Value>binary</Value>
        </Header>
        <Header>
            <Name>content-id</Name>
            <Value>&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;</Value>
            <Id asUrl="cid:0@4cd748bf-d5ea ...">0@4cd748bf-d5ea ...</Id>
        </Header>
        <Content encoding="base64">PD94bWwgdmVyc2 ...</Content>
    </Part>
</Multipart>
```

Output: Beispiel eines erzeugten MIME-Body-Dokuments

```
--7aed8c79-1491-41a3-9122-c82b15d88fd0
content-type: application/xop+xml; charset=UTF-8; type="text/xml"
content-transfer-encoding: binary
content-id: <0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d>


<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:stoc="http://example.com/stockquote.xsd">
    <soapenv:Header/>
    <soapenv:Body>
        <blub>SGFsbG8gSGFsbG8g ... </blub>
        <bla>SGFsbG8=</bla>
        <foo>SGFsbG8=</foo>
        <bar>SGFsbG8=</bar>
    </soapenv:Body>
</soapenv:Envelope>
--7aed8c79-1491-41a3-9122-c82b15d88fd0--
```

10.8.2.8 Operation FlattenXml

Mit der Adapter-Operation `FlattenXml` lässt sich ein MIME-Multipart-XML-Dokument erzeugen, das für den Versand mit einem Managed File Transfer-Adapter der X4 BPMS vorbereitet ist, z.B. mit dem *HTTP(S) Adapter*.

Das Ergebnis-XML-Dokument ist hierfür wie folgt aufgebaut:

- Es enthält einen MIME-Body als Base64-kodierte Zeichenkette.
- Die Header sind als XML-Struktur verfügbar, um einen X4 Transfer-Adapter einfach konfigurieren können.
- Das erzeugte Dokument liegt im MIME-XML-Format vom Typ *Part* vor. Sie können es daher verwenden, um weitere Multipart-Dokumente zu erzeugen.

Anwendungsfall: Webservice-Aufrufe, bei denen Multipart-Dokumente benötigt werden

Input: Beispiel eines MIME-XML-Dokuments

```
<Multipart>
    <Header>
        <Name>content-transfer-encoding</Name>
        <Value>binary</Value>
    </Header>
    <Header>
        <Name>content-type</Name>
        <Value>multipart/related; type="application/xop+xml";
            start="&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;";
            start-info="text/xml"</Value>
        <Type>multipart/related</Type>
        <Parameter>
            <Name>type</Name>
            <Value>application/xop+xml</Value>
        </Parameter>
        <Parameter>
            <Name>start</Name>
            <Value>&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;</Value>
        </Parameter>
        <Parameter>
            <Name>start-info</Name>
            <Value>text/xml</Value>
        </Parameter>
    </Header>
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>application/xop+xml; charset=UTF-8; type="text/xml"</Value>
            <Type>application/xop+xml</Type>
            <Parameter>
                <Name>charset</Name>
                <Value>UTF-8</Value>
            </Parameter>
            <Parameter>
                <Name>type</Name>
                <Value>text/xml</Value>
            </Parameter>
        </Header>
        <Header>
            <Name>content-transfer-encoding</Name>
            <Value>binary</Value>
        </Header>
        <Header>
            <Name>content-id</Name>
            <Value>&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;</Value>
            <Id asUrl="cid:0@4cd748bf-d5ea ...">0@4cd748bf-d5ea ...</Id>
        </Header>
        <Content encoding="base64">PD94bWwgdmVyc2 ...</Content>
    </Part>
</Multipart>
```

Output: Beispiel eines erzeugten MIME-XML-Dokuments

```
<Part>
    <Header>
        <Name>content-transfer-encoding</Name>
        <Value>binary</Value>
    </Header>
    <Header>
        <Name>content-type</Name>
        <Value>multipart/related; type="application/xop+xml";
          start="&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;";
          start-info="text/xml"; boundary=da9a6ade-4452-4546-851c-84da7b638c24</Value>
        <Type>multipart/related</Type>
        <Parameter>
            <Name>type</Name>
            <Value>application/xop+xml</Value>
        </Parameter>
        <Parameter>
            <Name>start</Name>
            <Value>&lt;0@4cd748bf-d5ea-4d2f-89cd-18eb54442e0d&gt;</Value>
        </Parameter>
        <Parameter>
            <Name>start-info</Name>
            <Value>text/xml</Value>
        </Parameter>
        <Parameter>
            <Name>boundary</Name>
            <Value>da9a6ade-4452-4546-851c-84da7b638c24</Value>
        </Parameter>
    </Header>
    <Content encoding="base64">LS1kYTlhNmFkZ ...</Content>
</Part>
```

10.8.2.9 Operation PartFromX4Document

Mit der Operation `PartFromX4Document` wird der Adapter-Input als Rumpf eines Mime-Dokuments konvertiert (im XML-Format dieses Adapters). Es wird ein `Content-Type` Header erzeugt mit den entsprechenden Werten aus dem X4-Dokument (diese Werte lassen sich mit dem Document Properties Changer beeinflussen).

X4Dokument-Eigenschaften umwandeln in den Content-Type Header

X4Document *mimeType*	X4Document *encoding*	resultierender `Content-Type` Header
leer	*leer*	*kein Header*
leer	`someCharset`	*kein Header*
`some/type`	*leer*	`Content-Type: some/type`
`some/type`	`someCharset`	`Content-Type: some/type; charset=someCharset`

Diese Operation ist vergleichbar mit folgender Adapterfolge:

1. Base64 (encode) – Input als Base64-String verpacken
2. ElementWrapper (wrap) – Base64-String in ein XML-Dokument verpacken
3. XSL – XML-Dokument in die passende Form bringen (und die Header erzeugen)

10.8.3 MIME Multipart Reader

`MIME Multipart Reader`: Liest Bodyparts einer Multipart-Message aus und gibt einen oder alle Bodyparts als Text oder als Base64-kodiertes XML-Dokument mit Header-Informationen aus.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`GetDocument:`* Liest ein bestimmtes Teil der Multipart-Message aus, das im Parameter `index` angegeben wurde. • *`Traverse`*: Liest mithilfe eines `Condition`-Bausteins alle Teile der Multipart-Message in einer Schleife nacheinander ein. Sobald alle Teile eingelesen wurden, liefert der Adapter den Status `0` zurück.

Parameter

`index`	Index (Position) des Teils, das bei der Operation `GetDocument` eingelesen werden soll. **Mögliche Werte:** • Beliebige natürliche Zahl (0 = Standard)
`toXML`	Alle Header-Informationen der Multipart-Message als Elemente und den Inhalt des gelesenen Teils Base64-kodiert in einem XML-Dokument ausgeben. **Mögliche Werte:** • *`ja`*: XML-Dokument mit Header-Informationen und Bodypart ausgeben • *`nein`*: Gelesene Multipart-Message als Text ausgeben (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.8.3.1 Beispiel

Der Adapter erwartet eine Multipart-Message als Input:

```
From: John Doe <example@example.com>
MIME-Version: 1.0
Content-Type: multipart/mixed;
        boundary="XXXXboundary text"

This is a multipart message in MIME format.

--XXXXboundary text
Content-Type: text/plain

This is the body text at index 0

--XXXXboundary text
Content-Type: text/plain;
Content-Disposition: attachment;
        filename="test.txt"

This is the body text at index 1

--XXXXboundary text--
```

10.8.3.2 GetDocument

Mit der Operation `GetDocument` liest der Adapter das nachfolgende Teil ein, wenn der Parameter `index` auf `0` und `toXML` auf `ja` gesetzt wurden:

```
This is the body text at index 0
```

10.8.3.3 Traverse

Die Operation `Traverse` liest mithilfe eines `Condition`-Bausteins alle Teile der Multipart-Message in einer Schleife nacheinander ein. Das nachfolgende Beispiel besitzt folgende Parameterkonfiguration: `index` = `0` und `toXML` = `ja`.

Im 1. Schleifendurchlauf wird das erste Teil (Header-Informationen und Bodypart) als Base64-kodiertes Element eingelesen:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
    <headers>
        <header>
            <name>Content-Type</name>
            <value>text/plain</value>
        </header>
    </headers>
    <content encoding="base64">VGhpcyBpcyB0aGUgYm9keSB0ZXh0IGF0IGluZGV4IDANCg==</
content>
</document>
```

Im 2. Schleifendurchlauf wird das zweite Teil (Header-Informationen und Bodypart) als Base64-kodiertes Element eingelesen:

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
    <headers>
        <header>
            <name>Content-Type</name>
            <value>text/plain;</value>
        </header>
        <header>
            <name>Content-Disposition</name>
            <value>attachment;
                filename="test.txt"
            </value>
        </header>
    </headers>
    <content encoding="base64">VGhpcyBpcyB0aGUgYm9keSB0ZXh0IGF0IGluZGV4IDENCg==</
content>
</document>
```

10.8.4 POP/SMTP

`POP/SMTP`: Sendet und empfängt E-Mail-Nachrichten über SMTP (Simple Mails Transfer Protocol) bzw. POP3 (Post Office Protocol v3).

Der Adapter unterstützt Dateianhänge und MIME-Typ-Angaben für den E-Mail-Body u.a. zum Versenden von HTML-E-Mails. Zudem wird das Senden und Empfangen von S/MIME-verschlüsselten und/oder -signierten E-Mail-Nachrichten unterstützt.

Eigenschaften (Properties)

`I/O Mode (input_type)`	Art der Quelle der zu verarbeitenden Daten **Mögliche Werte:** • *`Actions`*: Daten aus dem vorangegangenen Prozess-Schritt verarbeiten • *`Directory`*: Alle Dateien eines angegebenen Ordners als Anhang verarbeiten (oder Dateien gemäß Filterregel in Parameter filter)

`File name (documentname)`	Name des zu sendenden Dateianhangs (wenn `I/O Mode` auf *`Actions`* gesetzt ist).
	
	ⓘ Wenn diese Eigenschaft dynamisch parametriert werden soll, muss im zugehörigen XSL-Mapping der technische Name verwendet werden. Dieser Name kann abweichen und ist in Klammern angegeben.
	
	Beispiel
	
	`<xsl:processing-instruction name="DynamicParameter">`
	`documentname="`
	`<xsl:value-of select="."/>`
	`"`
	`</xsl:processing-instruction>`
	
	Mögliche Werte:
	
	• Beliebiger Dateiname (z.B. *`Data.xml`*)
	• Mehrere Dateinamen, durch Semikolon getrennt (wenn Sie dem Transfer-Adapter über Fork/Join mehrere Input-Dokumente übergeben, dann können auch mehrere Dateien gemäß der Priorität der Verzweigungen nacheinander verarbeitet werden (z.B. *`File1.xml;File2.xml`*)

`Delete (delete)`	E-Mail nach erfolgreichem Empfang vom POP3-Server löschen (wenn Eigenschaft `Operation` auf *`Receive`* gesetzt ist)
`Operation (action)`	Bestimmt, welche Operation mit den verarbeiteten Dateien durchgeführt wird. Je nach Wert in `I/O Mode` unterscheidet sich die Funktionsweise des Adapters. **Mögliche Werte:** • *`Send`*: E-Mail über den SMTP-Server senden • *`Receive`*: E-Mail vom POP3-Server empfangen

Parameter

`from`	E-Mail-Adresse des Absenders; Bei Bedarf mit Angabe einer Anzeigenamens **Mögliche Werte:** Beliebige gültige E-Mail-Adresse (z.B. *`name@domain.tld (SampleName)`*)
`sender`	Technischer Absender (falls abweichend von `from`), der die Nachricht verschickt; Bei Bedarf mit Angabe einer Anzeigenamens **Mögliche Werte:** Beliebige gültige E-Mail-Adresse (z.B. *`name@domain.tld (SampleName)`*)
`to`	E-Mail-Adresse(n), an die die E-Mail gesendet werden soll; Bei Bedarf mit Angabe einer Anzeigenamens **Mögliche Werte:** Beliebige gültige E-Mail-Adresse, bei mehreren Adressen durch Semikolon getrennt (z.B. *`name@domain.tld;email@host.tld (SampleName)`*)
`cc`	E-Mail-Adresse(n), an die die E-Mail als Kopie gesendet werden soll; Bei Bedarf mit Angabe einer Anzeigenamens **Mögliche Werte:** Beliebige gültige E-Mail-Adresse, bei mehreren Adressen durch Semikolon getrennt (z.B. *`name@domain.tld (SampleName);email@host.tld`*)
`bcc`	E-Mail-Adresse(n), an die die E-Mail als Blindkopie gesendet werden soll; Bei Bedarf mit Angabe einer Anzeigenamens **Mögliche Werte:** Beliebige gültige E-Mail-Adresse, bei mehreren Adressen durch Semikolon getrennt (z.B. *`name@domain.tld;email@host.tld (SampleName)`*)
`replyTo`	E-Mail-Adresse, an die eine Antwort verschickt werden soll **Mögliche Werte:** Beliebige gültige E-Mail-Adresse (z.B. *`name@domain.tld`*)
`subject`	Betreff der E-Mail
`body`	Inhalt der E-Mail (wenn die Input-Daten des Adapters ein HTML-Dokument ist, so wird dieses als HTML-Nachrichten-Body verwendet) **Mögliche Werte:** Beliebige Zeichenkette
`mimeTypeBody`	MIME-Typ des E-Mail-Inhalts (ermöglicht die Darstellung von HTML-E-Mails oder mehreren Anhängen) **Mögliche Werte:** Gültige MIME-Typen (z.B. *`multipart/mixed`*)

`charset`	Zu verwendender Zeichensatz des E-Mail-Inhalts **Mögliche Werte:** Gültige Zeichensatz-Angabe (z.B. *`ISO-8859-15`*)
`smtp`	IP-Adresse oder Host-Name des SMTP-Servers zum Senden von E-Mails **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *`smtp.domain.tld`*)
`smtpPort`	Port des SMTP-Servers zum Senden von E-Mails **Mögliche Werte:** Beliebige positive ganze Zahl, z.B. *250*
`smtpSecurityProtocol`	Zu verwendendes SMTP-Sicherheitsprotokoll **Mögliche Werte:** • `NONE`: Kein Protokoll verwenden • `TLS`: TLS als Sicherheitsprotokoll verwenden • `SSL`: SSL als Sicherheitsprotokoll verwenden
`smtpAllowAllHosts`	**Alle** Hosts für SMTP erlauben ⚠ Bitte beachten Sie, dass auch Hosts erlaubt werden, die kein gültiges Zertifikat haben. ℹ Dieser Parameter ist nur gültig, wenn SSL als Sicherheitsprotokoll verwendet wird.
`pop3`	IP-Adresse oder Host-Name des POP3-Servers zum Empfangen von E-Mails **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *`pop3.domain.tld`*)
`pop3Port`	Port des POP3-Servers zum Empfangen von E-Mails **Mögliche Werte:** Beliebige positive ganze Zahl
`pop3SecurityProtocol`	Zu verwendendes POP3-Sicherheitsprotokoll **Mögliche Werte:** • `NONE`: Kein Protokoll verwenden • `TLS`: TLS als Sicherheitsprotokoll verwenden • `SSL`: SSL als Sicherheitsprotokoll verwenden

`pop3AllowAllHosts`	**Alle** Hosts für POP3 erlauben ⚠ Bitte beachten Sie, dass auch Hosts erlaubt werden, die kein gültiges Zertifikat haben. ℹ Dieser Parameter ist nur gültig, wenn SSL als Sicherheitsprotokoll verwendet wird.
`login`	Benutzername für den POP3-Server (beim Empfangen einer E-Mail) oder für den SMTP-Server (beim Senden einer E-Mail) **Mögliche Werte:** Beliebige Zeichenkette (z.B. *`name@domain.tld`*)
`password`	Passwort für den POP3-Server (beim Empfangen einer E-Mail) oder für den SMTP-Server (beim Senden einer E-Mail)
`oAuth2`	Legt fest, ob das OAuth 2.0-Protokoll zum Log-In verwendet werden soll. **Mögliche Werte:** • *ja*: `oAuth2` wird zum Log-in verwendet • *nein*: `oAuth2` wird nicht zum Log-in verwendet ⚠ Wenn OAuth 2.0 verwendet werden soll, • muss die SSL-Verschlüsselung für die Verbindung zum POP-Server durch setzen des Parameters `ssl` aktiviert sein. • muss beim Parameter `password` der Access Token eingetragen werden.
`oAuth2_twoLineAuth`	Aktiviert das POP3-Protokoll für Microsoft 365. ⚠ Muss zusätzlich zu `oAuth2` aktiviert werden, um das POP3-Protokoll mit Microsoft 365 zu nutzen. **Mögliche Werte:** • *ja*: `oAuth2_twoLineAuth` wird zusätzlich zu `oAuth2` aktiviert • *nein*: `oAuth2_twoLineAuth` wird nicht zusätzlich zu `oAuth2` aktiviert

`file_limit`	Anzahl der zu sendenden Dateianhänge begrenzen (wenn `I/O Mode` auf *`Directory`* gesetzt ist und somit mehrere Dateien eines Ordners angehängt werden sollen) **Mögliche Werte:** • Beliebige positive ganze Zahl • *0*: Keine Dateien anhängen (Standard)
`dir`	Lokaler Ordner auf dem X4 Server, z.B. im X4 Repository, in dem Dateien liegen, die als Dateianhänge per E-Mail gesendet werden sollen (wenn `I/O Mode` auf *`Directory`* gesetzt ist) **Mögliche Werte:** Beliebige Zeichenkette (z.B. *`%USER_COLLECTION%\Folder`*)
`filter`	Zu versendenden Datei-Anhänge nach Dateinamen filtern über die Platzhalter *** und *?* **Mögliche Werte:** • Beliebige Zeichenkette, so gilt beispielsweise *`*.dat`* nur für Dateien mit der Dateiendung *`*.dat`* • ***: Alle Dateien verwenden (Standard)
`EML`	Nachricht im EML-Format verschicken (dies erfordert eine EML-Nachricht als Adapter-Input) oder Nachricht im EML-Format empfangen **Mögliche Werte:** • *`ja`*: Im EML-Format verschicken • *`nein`*: Standard-E-Mail-Format verwenden (Standard)
`base64`	Anhänge von Nachrichten (Multipart-Teile) immer Base64-kodieren (wenn die Eigenschaft `Operation` auf *`Receive`* gesetzt ist) **Mögliche Werte:** • *`ja`*: Alle Multipart-Teile Base64-kodieren • *`nein`*: Nicht Base64-kodieren (Standard)

`encrypt`	Nachricht mit S/MIME verschlüsseln (*Multipart/Encrypted*) **Mögliche Werte:** • *`ja`*: S/MIME-Verschlüsselung aktivieren (erfordert die Angabe zur Schlüssel-/Zertifkatsdatei in Parameter `Pkcs12` und der Passphrase in Parameter `passphrase`) • *`nein`*: Keine Verschlüsselung (Standard)
`sign`	Nachricht mit S/MIME signieren (*Multipart/Signed*) **Mögliche Werte:** • *`ja`*: S/MIME-Signatur hinzufügen (erfordert die Angabe zur Schlüssel-/Zertifikatsdatei in Parameter `Pkcs12` und der Passphrase in Parameter `passphrase`) • *`nein`*: Keine Signatur (Standard)
`Pkcs12`	Pfad zur einer gültigen Schlüssel-/Zertifikatsdatei im *PKCS#12*-Format **Mögliche Werte:** Beliebige gültige Pfadangabe im lokalen Dateisystem des *X4 Servers* (z.B. *`%USER_COLLECTION\Project\Folder\Key.p12`*)
`passphrase`	Passwort/Passphrase zur Schlüssel-/Zertifikatsdatei im *PKCS#12*-Format, die Sie in Parameter `Pkcs12` angegeben haben **Mögliche Werte:** Beliebige gültige Zeichenkette
`propertiesType`	Datenformat für eine Properties-Datei, in der weitere *JavaMail*-Properties hinterlegt werden können **Mögliche Werte:** • *`text`*: Properties-Datei in Textformat • *`xml`*: Properties-Datei in XML-Format, siehe http://www.ibm.com/developerworks/library/j-tiger02254/index.html

`propertiesUrl`	**Veralteter Parameter** Ab Version 6.2.0 der X4 BPMS kann das zu verwendende Sicherheitsprotokoll über die Parameter `smtpSecurityProtocol` und `pop3SecurityProtocol` gesetzt werden. Beim Verwenden des nun veralteten Parameters `propertiesURL` wird eine Warnung im Server-Log ausgegeben. URL zu einer Properties-Datei, in der weitere *JavaMail*-Properties hinterlegt werden können, siehe https://javamail.java.net/nonav/docs/api/com/sun/mail/smtp/package-summary.html. Die verknüpfte Datei muss im Format vorliegen, das im Parameter `propertiesType` angegeben wurde. Wird die `propertiesURL` gesetzt, werden die Parameter aus der Properties-Datei verwendet. Andernfalls werden die Adapter-Parameter verwendet. **Mögliche Werte:** Beliebige gültige URL, z.B. `xstore://Project/Folder/File.properties`

Welche E-Mail-Adressen formal gültig sind, kann den Richtlinien *RFC 5321* und *RFC 5322* entnommen werden.

10.9 ERP

- Abas ERP Connector
- SAP Connector

10.9.1 Abas ERP Connector

`Abas ERP Connector`: Stellt eine Verbindung zum Abas ERP-System über dessen *jedp*-Schnittstelle her und führt verschiedene Operationen aus.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Connector durchführt **Mögliche Werte:** • `Query`: Datensätze und Tabelleninhalte abfragen • `Info`: Metadaten von abas abfragen • `Write`: Datensätze und Tabelleninhalte hinzufügen oder ändern

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.abas.AbasAdapter*: Hauptklasse (Standard)
appName	Name der Applikation **Mögliche Werte:** Beliebige Zeichenkette
edpPort	EDP-Port **Mögliche Werte:** Beliebige ganze Zahl, z.B. *6550*
mandant	Name des Mandanten **Mögliche Werte:** Beliebige Zeichenkette, z.B. *systemtest*
host	IP-Adresse oder Host-Name des Servers **Mögliche Werte:** Beliebige gültige Zeichenkette ohne Protokollangabe, z.B. *example.com*
password	Passwort für die Authentifizierung **Mögliche Werte:** Beliebige Zeichenkette
edpVersion	EDP-Version **Mögliche Werte:** Beliebige ganze Zahl, z.B. *3.35* (Standard)
wait	Lock-Verhalten der Session **Mögliche Werte:** • `WAITINFINITE`: Warten, bis Lock freigegeben ist • `WAIT`: Warten, bis Lock freigegeben oder die Zeitbeschränkung erreicht ist • `NOWAIT`: Fehler ohne zu warten sofort zurückgeben (Standard)
timeout	Zeitbeschränkung für den Verbindungsaufbau in Sekunden **Mögliche Werte:** Beliebige ganze Zahl, z.B. *30* (standard)

Statuswerte

1 (OK)	Die Ausführung war erfolgreich

-1 (ERROR)	Bei der Ausführung kam es zu einem Fehler, z.B. wegen eines ungültigen XML-Inputs
0 (TIMEOUT)	Bei der Abfrage des abas ERP-Systems kam es zu einer Zeitüberschreitung

Input

Für die verschiedenen Operationen erwartet der Adapter folgende XML-Input-Strukturen:

Operation: Query

```
<Abas>
    <!-- Query ausführen -->
    <Query resulttag="...">
        <DatabaseGroup>00:01</DatabaseGroup>
        <Fields>
            <_ans/>
            <_str/>
            <_nummer/>
        </Fields>
        <Selection>nummer=20001!20002</Selection>
    </Query>
    <Query resulttag="...">
        <DatabaseGroup>01:01</DatabaseGroup>
        <Fields>
            <_schlag/>
            <_yvertragsnr/>
            <_kunde/>
        </Fields>
        <Selection>kunde=20001!20002</Selection>
    </Query>
</Abas>
```

Operation: Info

```
<Abas>
    <!-- Aktion ausführen -->
    <Info resulttag="...">
        <Nummer>1000</Nummer>
        <HeaderFields>
            <_bstart>1</_bstart>
        </HeaderFields>
        <TableFields>
            <_tysuch/>
            <_tyname/>
        </TableFields>
    </Info>
    <Info resulttag="...">
        <Nummer>1000</Nummer>
        <HeaderFields>
            <_bstart>1</_bstart>
        </HeaderFields>
        <TableFields>
            <_tysuch/>
        </TableFields>
    </Info>
</Abas>
```

Operation: Write

```
<Abas>
    <!-- Bestehenden Datensatz ändern -->
    <Update resulttag="...">
        <Database>00</Database>
        <Group>01</Group>
        <Nummer>20001</Nummer>
        <HeaderFields>
            <_str>...</_str>
        </HeaderFields>
    </Update>
    <!-- Neuer Datensatz -->
    <Insert resulttag="...">
        <Database>00</Database>
        <Group>01</Group>
        <HeaderFields>
            <_nummer>200171</_nummer>
            <_such>Testsuch</_such>
            <_ans2>Testans</_ans2>
            <_str2>Testgasse</_str2>
            <_plz2>12345</_plz2>
            <_nort2>Testhausen</_nort2>
            <_zbed>100</_zbed>
        </HeaderFields>
    </Insert>
    <!-- Zeile in bestehendem Datensatz ändern -->
    <Update resulttag="...">
        <Database>71</Database>
        <Group>02</Group>
        <Nummer>4</Nummer>
        <Table>
            <Row nr="2">
                <TableFields>
                    <_ytg02beschreibung>blau</_ytg02beschreibung>
                </TableFields>
            </Row>
        </Table>
    </Update>
    <!-- Neue Zeile in bestehendem Datensatz -->
    <Update resulttag="...">
        <Database>71</Database>
        <Group>02</Group>
        <Nummer>4</Nummer>
        <Table>
            <Row nr="2" new="true">
                <TableFields>
                    <_ytg02beschreibung>gelb</_ytg02beschreibung>
                </TableFields>
            </Row>
        </Table>
    </Update>
</Abas>
```

Ausgabe

Für die verschiedenen Operationen gibt der Adapter folgende XML-Ergebnisdokumente aus:

Operation: Query

```
<Result>
    <Adresse>
        <_ans>Max Mustermann</_ans>
        <_str>Musterallee 2</_str>
        <_nummer>20001</_nummer>
    </Adresse>
    <Adresse>
        <_ans>Manfred Mustermann</_ans>
        <_str>Testweg 1</_str>
        <_nummer>20002</_nummer>
    </Adresse>
    <Vertrag>
        <_schlag>RECHNUNG.2017.10</_schlag>
        <_yvertragsnr>01-000001-001</_yvertragsnr>
        <_kunde>123</_kunde>
    </Vertrag>
   <!-- ggf. weitere Knoten -->
</Result>
```

Operation: Info

```
<Result>
    <Test>
        <_tysuch>XY123</_tysuch>
        <_tyname>Test Abas-Server</_tyname>
    </Test>
    <Test2>
        <_tysuch>XY123</_tysuch>
    </Test2>
</Result>
```

Operation Write

```
<Result>
    <Testupdate1>updated</Testupdate1>
    <Testinsert1>inserted</Testinsert1>
    <Testupdate2>updated</Testupdate2>
    <Testupdate3>updated</Testupdate3>
</Result>
```

Abhängigkeiten

Der Connector verwendet die Bibliothek *jedp* von abas, `jedp-3.3.5.jar`.

Die aktuell unterstütze Version ist 3.3.5.
Die Bibliothek wird aus lizenzrechtlichen Gründen nicht ausgeliefert. Zur Verwendung diese Bibliothek im X4 Server-Installationsordner in Ordner `wildfly\modules\system\layers\base\de\softproject\x4\extensions\main\` kopieren, gemeinsam mit einer Datei `module.xml` mit folgendem Inhalt:

module.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="de.softproject.x4.extensions">
    <resources>
        <resource-root path="jedp-3.3.5.jar" />
    </resources>
</module>
```

10.9.2 SAP Connector

10.9.2.1 IDoc-Strukturen

Wie IDocs im Raw-Format und XML-IDocs aufgebaut sind

10.9.2.1.1 IDoc im Raw-Format

Ein *IDoc Raw* -Dokument ist ein XML-Dokument, das neben Metadaten zur Nachricht und Strukturinformationen ein SAP *IDoc* im Rohformat als Text enthält. Das XML-Dokument ist nach folgendem Muster aufgebaut:

```
<X4SAP TID="<!--Transaktions-ID-->">
  <X4IDoc type="<!--IDoc-Typ-->" description="">
    <X4Mandatory>
      <X4RecipientPort><!--Empfänger-Port--></X4RecipientPort>
      <X4RecipientPartnerType><!--Partnertyp-Kürzel--></X4RecipientPartnerType>
      <X4RecipientPartnerNumber><!--Partnernummer--></X4RecipientPartnerNumber>
      <X4RecipientPartnerFunction />
      <X4RecipientAdress />
      <X4RecipientLogicalAdress />
      <X4SenderPort><!--Sender-Port--></X4SenderPort>
      <X4SenderPartnerType><!--Partnertyp-Kürzel--></X4SenderPartnerType>
      <X4SenderPartnerNumber><!--Partnernummer--></X4SenderPartnerNumber>
      <X4SenderPartnerFunction />
      <X4SenderAddress />
      <X4SenderLogicalAddress />
      <MessageType>
        <MSGTYPE><!--Nachrichtentyp--></MSGTYPE>
      </MessageType>
    </X4Mandatory>
    <X4Structure>
      <Tabellenname SEGMENT="1">
        <!--Segment 1 (Steuersatz)-->
      </Tabellenname>
    </X4Structure>
    <RAW>
       <!--Rohdaten der IDoc-Nachricht (Text)-->
    </RAW>
  </X4IDoc>
</X4SAP>
```

10.9.2.1.2 IDoc im XML-Format

Ein IDoc-XML-Dokument enthält eine IDoc-Nachricht, die komplett in XML-Elemente umgewandelt wurde. Zudem enthält das XML-Dokument Metadaten zur Nachricht und Strukturinformationen. Innerhalb des Elements `<IDOC>` werden sämtliche Tabellen hintereinander hinterlegt.

Ein IDoc im XML-Format ist nach folgendem Muster aufgebaut:

```
<X4SAP TID="<!--Transaktions-ID-->">
  <X4IDoc type="<!--IDoc-Typ-->" description="">
    <X4Mandatory>
      <X4RecipientPort><!--Empfänger-Port--></X4RecipientPort>
      <X4RecipientPartnerType><!--Partnertyp-Kürzel--></X4RecipientPartnerType>
      <X4RecipientPartnerNumber><!--Partnernummer--></X4RecipientPartnerNumber>
      <X4RecipientPartnerFunction />
      <X4RecipientAdress />
      <X4RecipientLogicalAdress />
      <X4SenderPort><!--Sender-Port--></X4SenderPort>
      <X4SenderPartnerType><!--Partnertyp-Kürzel--></X4SenderPartnerType>
      <X4SenderPartnerNumber><!--Partnernummer--></X4SenderPartnerNumber>
      <X4SenderPartnerFunction />
      <X4SenderAddress />
      <X4SenderLogicalAddress />
      <MessageType>
        <MSGTYPE><!--Nachrichtentyp--></MSGTYPE>
      </MessageType>
    </X4Mandatory>
    <X4Structure>
      <Tabellenname SEGMENT="1">
        <!--Segment 1 (Steuersatz)-->
      </Tabellenname>
    </X4Structure>
    <RAW>
       <!--Rohdaten der IDoc-Nachricht (Text)-->
    </RAW>
  </X4IDoc>
</X4SAP>
```

10.9.2.2 SAP Connector (JCo 2)

`SAP Connector (JCo 2)`: Ermöglicht die Übermittlung von Daten vom X4 ESB an ein *SAP R/3*-System. Er ruft einen beliebigen Remote Function Call (RFC) bzw. einen beliebigen BAPI-Funktionsbaustein in einem SAP-System auf. Zudem können IDocs an *SAP R/3* versandt werden.

Ehemalige Bezeichnung bis Version 4.5.1: *SAP (JCo 2) Request Adapter*

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *Execute*: Funktions-Adapter ausführen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.sapadapter.SAPAdapter*: Hauptklasse (Standard)
x4.connectionPoolEnabled	Connection Pooling verwenden
x4.connectionPoolUID	ID des Connection Pools. Der SAP-Adapter kann einen bereits existierenden Connection Pool verwenden, wenn dieselben Verbindungsparameter und dieselbe ID des Connection Pools gesetzt wurden.
x4.connectionPoolMax	Maximale Anzahl definierter Verbindungen im Connection Pool
x4.transactionEnabled	Transaktionale Verarbeitung aktivieren oder deaktivieren.
x4.transactionID	Transaktions-ID; alle Operationen des SAP-Adapters innerhalb eines Prozesses werden als eine Transaktion ausgeführt und besitzen dieselbe Transaktions-ID. Der SAP-Adapter kann eine bereits gestartete Transaktion verwenden, wenn sie dieselben Transaktionsparameter und dieselbe Transaktions-ID besitzen.
jco.client.client	SAP-spezifischer Parameter: SAP-Mandant
jco.client.user	SAP-spezifischer Parameter: Benutzername
jco.client.passwd	SAP-spezifischer Parameter: Passwort
jco.client.lang	SAP-spezifischer Parameter: Anmeldesprache
jco.client.sysnr	SAP-spezifischer Parameter: SAP-System-Nummer
jco.client.ashost	SAP-spezifischer Parameter: SAP-Applikations-Server
jco.client.mshost	SAP-spezifischer Parameter: SAP-Nachrichten-Server
jco.client.gwhost	SAP-spezifischer Parameter: Gateway Host
jco.client.gwserv	SAP-spezifischer Parameter: Gateway Service
jco.client.r3name	SAP-spezifischer Parameter: R/3-Name
jco.client.group	SAP-spezifischer Parameter: Gruppe der SAP-Applikations-Server

jco.client.tpname	SAP-spezifischer Parameter: Programm-ID einer externen Server-Anwendung
jco.client.tphost	SAP-spezifischer Parameter: Host der externen Server-Anwendung
jco.client.type	SAP-spezifischer Parameter: Art des entfernten Hosts **Mögliche Werte:** • *2*: R/2 • *3*: R/3 • *E*: Extern
jco.client.trace	SAP-spezifischer Parameter: RFC-Trace **Mögliche Werte:** • *0*: RFC-Trace deaktivieren • *1*: RFC-Trace aktivieren
jco.client.codepage	SAP-spezifischer Parameter: Anfängliche Codepage in SAP-Notation
jco.client.abap_debug	SAP-spezifischer Parameter: ABAP-Debugging **Mögliche Werte:** • *0*: ABAP-Debugging deaktivieren • *1*: ABAP-Debugging aktivieren
jco.client.use_sapgui	SAP-spezifischer Parameter: Name einer grafischen SAP-Benutzeroberfläche, die verwendet werden soll
jco.client.getsso2	SAP-spezifischer Parameter: SSO-Ticket nach der Anmeldung erhalten **Mögliche Werte:** • *0*: Kein SSO-Ticket erhalten • *1*: SSO-Ticket nach der Anmeldung erhalten
jco.client.mysapsso2	SAP-spezifischer Parameter: Name des SAP-Cookies Version 2, der als Anmeldungs-Ticket verwendet werden soll

jco.client.x509cert	SAP-spezifischer Parameter: Name des X509-Zertifikats, das als Anmeldungs-Ticket verwendet werden soll
jco.client.lcheck	SAP-spezifischer Parameter: Anmelde-Überprüfung beim Öffnen **Mögliche Werte:** • *0*: Anmelde-Überprüfung deaktivieren • *1*: Anmelde-Überprüfung beim Öffnen aktivieren
jco.client.grt_data	SAP-spezifischer Parameter: Zusätzliche Daten für die grafische SAP-Benutzeroberfläche
jco.client.use_guihost	SAP-spezifischer Parameter: Host, zu dem die entfernte grafische SAP-Benutzeroberfläche weitergeleitet werden soll
jco.client.use_guiserv	SAP-spezifischer Parameter: Dienst, zu dem die entfernte grafische SAP-Benutzeroberfläche weitergeleitet werden soll
jco.client.use_guiprogid	SAP-spezifischer Parameter: Programm-ID des Servers, der die entfernte grafische SAP-Benutzeroberfläche startet
jco.client.snc_mode	SAP-spezifischer Parameter: *Secure network connection* (SNC)-Modus **Mögliche Werte:** • *0*: SNC-Modus deaktivieren • *1*: SNC-Modus aktivieren
jco.client.snc_partnername	SAP-spezifischer Parameter: SNC-Partner, z. B. *p:CN=R3, O=XYZ-INC, C=EN*
jco.client.snc_qop	SAP-spezifischer Parameter: SNC-Sicherheits-Level **Mögliche Werte:** *1…9*: SNC-Sicherheits-Level
jco.client.snc_myname	SAP-spezifischer Parameter: SNC-Name (überschreibt den Standard-SNC-Partner)
jco.client.snc_lib	SAP-spezifischer Parameter: Pfad zu einer Bibliothek, die den SNC-Dienst bereitstellt
jco.client.dest	SAP-spezifischer Parameter: R/2-Ziel (destination)

jco.client.saplogon_id	SAP-spezifischer Parameter
jco.client.extiddata	SAP-spezifischer Parameter: Daten zur externen Authentifizierung (PAS)
jco.client.extidtype	SAP-spezifischer Parameter: Art der externen Authentifizierung (PAS)

ⓘ Eine vollständige Beschreibung aller (Verbindungs-) Parameter finden Sie in der *SAP JCo 2*-Dokumentation.

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.9.2.2.1 RFC/BAPI aufrufen

Input

Der Adapter erwartet ein Input-XML-Dokument nach folgendem Muster, das den entsprechenden Funktionsbaustein ggf. mit Inputdaten und den betroffenen Tabellen aufruft:

```
<X4SAP>
 <bapicall name="BAPI-Name" id="IDoc-ID">
   <input>
      <INPUT>
         <!--Parameter und Parameterwerte -->
      </INPUT>
   </input>
   <table>
      <!--Tabellen-->
   </table>
 </bapicall>
</X4SAP>
```

Beispiel

Um BAPI `IDOCTYPE_READ_COMPLETE` in SAP aufzurufen, erwartet dieser Adapter folgendes Input-XML-Dokument:

```
<X4SAP>
 <bapicall name="IDOCTYPE_READ_COMPLETE" id="IDoc-ID">
   <input>
      <INPUT>
         <PI_CIMTYP />
         <PI_IDOCTYP> <!--IDoc-Typ--> </PI_IDOCTYP>
         <PI_RELEASE>SY-S</PI_RELEASE>
         <PI_VERSION>3</PI_VERSION>
      </INPUT>
   </input>
   <table>
      <TABLES>
         <PT_FIELDS>
            <SEGMENTTYP />
            <FIELDNAME />
            <INTLEN />
            <EXTLEN />
            <FIELD_POS />
            <BYTE_FIRST />
            <BYTE_LAST />
            <ROLLNAME />
            <DOMNAME />
            <DATATYPE />
            <DESCRP />
            <ISOCODE />
            <VALUETAB />
         </PT_FIELDS>
         <PT_FVALUES>
            <STRNAME />
            <FIELDNAME />
            <FLDVALUE_L />
            <FLDVALUE_H />
            <DESCRP />
         </PT_FVALUES>
         <PT_MESSAGES>
            <MESTYP />
            <DESCRP />
            <IDOCTYP />
            <CIMTYP />
            <RELEASED />
         </PT_MESSAGES>
         <PT_SEGMENTS>
            <NR />
            <SEGMENTTYP />
            <SEGMENTDEF />
            <QUALIFIER />
            <SEGLEN />
            <PARSEG />
            <PARPNO />
            <PARFLG />
            <MUSTFL />
            <OCCMIN />
            <OCCMAX />
            <HLEVEL />
```

```
            <DESCRP />
            <GRP_MUSTFL />
            <GRP_OCCMIN />
            <GRP_OCCMAX />
            <REFSEGTYP />
         </PT_SEGMENTS>
      </TABLES>
   </table>
 </bapicall>
</X4SAP>
```

10.9.2.2.2 IDoc senden

Input

Um ein IDoc an SAP zu senden, ein XML-Dokument wie das folgende als Input für den Adapter verwenden:

```
<X4SAP>
   <X4IDoc type="MATMAS02" description="">
      <X4Mandatory>
         <X4RecipientPartnerType>LS</X4RecipientPartnerType>
         <X4RecipientPartnerNumber>X4_T</X4RecipientPartnerNumber>
         <X4RecipientPartnerFunction />
         <X4SenderPort>SAPT01</X4SenderPort>
         <X4SenderPartnerType>LS</X4SenderPartnerType>
         <X4SenderPartnerNumber>ST01_200</X4SenderPartnerNumber>
         <X4SenderPartnerFunction />
         <X4SenderAddress />
         <X4SenderLogicalAddress />
         <X4MsgType>
            <MSGTYPE>MATMAS02</MSGTYPE>
         </X4MsgType>
      </X4Mandatory>
      <X4Structure>
<MATMAS02>
    <IDOC BEGIN="1">
        <EDI_DC40 SEGMENT="1">
            <TABNAM>EDI_DC40</TABNAM>
            <MANDT/>
            <DOCNUM/>
            <DOCREL/>
            <STATUS/>
            <DIRECT/>
            <OUTMOD/>
            <EXPRSS/>
            <TEST/>
            <IDOCTYP>MATMAS02</IDOCTYP>
            <CIMTYP/>
            <MESTYP>MATMAS</MESTYP>
            <MESCOD/>
            <MESFCT/>
            <STD/>
            <STDVRS/>
            <STDMES/>
            <SNDPOR>SAPJCOIDOC</SNDPOR>
            <SNDPRT>LS</SNDPRT>
            <SNDPFC/>
            <SNDPRN>JCOCLNT000</SNDPRN>
            <SNDSAD/>
            <SNDLAD/>
            <RCVPOR/>
            <RCVPRT>LS</RCVPRT>
            <RCVPFC/>
            <RCVPRN>TSTCLNT000</RCVPRN>
            <RCVSAD/>
            <RCVLAD/>
            <CREDAT>20050519</CREDAT>
            <CRETIM>104537</CRETIM>
            <REFINT/>
            <REFGRP/>
            <REFMES/>
```

```
            <ARCKEY/>
            <SERIAL/>
        </EDI_DC40>
        <E1MARAM SEGMENT="1">
            <MSGFN>005</MSGFN>
            <MATNR>BOXCOOKIES</MATNR>
            <ERSDA>20020801</ERSDA>
            <ERNAM>TIGGER</ERNAM>
            <LAEDA/>
            <AENAM/>
            <PSTAT>KBG</PSTAT>
            <LVORM/>
            <MTART>FERT</MTART>
            <MBRSH>L</MBRSH>
            <MATKL>G1113</MATKL>
            <BISMT/>
            <MEINS>PCE</MEINS>
            <BSTME/>
            <ZEINR/>
            <ZEIAR/>
            <ZEIVR/>
            <ZEIFO/>
            <AESZN/>
            <BLATT/>
            <BLANZ>000</BLANZ>
            <FERTH/>
            <FORMT/>
            <GROES/>
            <WRKST/>
            <NORMT/>
            <LABOR/>
            <EKWSL/>
            <BRGEW>0.550</BRGEW>
            <NTGEW>0.000</NTGEW>
            <GEWEI>KGM</GEWEI>
            <VOLUM/>
            <VOLEH/>
            <BEHVO/>
            <RAUBE/>
            <TEMPB/>
            <TRAGR/>
            <STOFF/>
            <SPART/>
            <KUNNR/>
            <WESCH/>
            <BWVOR/>
            <BWSCL/>
            <SAISO/>
            <ETIAR/>
            <ETIFO/>
            <EAN11/>
            <NUMTP/>
            <LAENG/>
            <BREIT/>
            <HOEHE/>
```

```
            <MEABM/>
            <PRDHA/>
            <CADKZ/>
            <ERGEW/>
            <ERGEI/>
            <ERVOL/>
            <ERVOE/>
            <GEWTO/>
            <VOLTO/>
            <VABME/>
            <KZKFG/>
            <XCHPF/>
            <VHART/>
            <FUELG/>
            <STFAK/>
            <MAGRV/>
            <BEGRU/>
            <QMPUR/>
            <RBNRM/>
            <MHDRZ/>
            <MHDHB/>
            <MHDLP/>
            <VPSTA>KBG</VPSTA>
            <EXTWG/>
            <MSTAE/>
            <MSTAV/>
            <MSTDE/>
            <MSTDV/>
            <KZUMW/>
            <KOSCH/>
            <NRFHG/>
            <MFRPN/>
            <MFRNR/>
            <BMATN/>
            <MPROF/>
            <PROFL/>
            <IHIVI/>
            <ILOOS/>
            <KZGVH/>
            <XGCHP/>
            <COMPL/>
            <KZEFF/>
            <RDMHD/>
            <IPRKZ/>
            <PRZUS/>
            <MTPOS_MARA/>
            <GEWTO_NEW/>
            <VOLTO_NEW/>
            <WRKST_NEW/>
            <AENNR/>
            <MATFI/>
            <CMREL/>
            <SATNR/>
            <SLED_BBD/>
            <GTIN_VARIANT/>
```

```
            <GENNR/>
            <SERLV/>
            <RMATP/>
            <E1MAKTM SEGMENT="1">
                <MSGFN>005</MSGFN>
                <SPRAS>D</SPRAS>
                <MAKTX>Schachtel mit Keksen</MAKTX>
                <SPRAS_ISO>DE</SPRAS_ISO>
            </E1MAKTM>
            <E1MAKTM SEGMENT="1">
                <MSGFN>005</MSGFN>
                <SPRAS>E</SPRAS>
                <MAKTX>Box of cookies</MAKTX>
                <SPRAS_ISO>EN</SPRAS_ISO>
            </E1MAKTM>
            <E1MARCM SEGMENT="1">
                <MSGFN>005</MSGFN>
                <WERKS>0001</WERKS>
                <PSTAT>BG</PSTAT>
                <LVORM/>
                <BWTTY/>
                <MAABC/>
                <KZKRI/>
                <EKGRP/>
                <AUSME/>
                <DISPR/>
                <DISMM/>
                <DISPO/>
                <PLIFZ>0</PLIFZ>
                <WEBAZ>0</WEBAZ>
                <PERKZ>M</PERKZ>
                <AUSSS>0.00</AUSSS>
                <DISLS/>
                <BESKZ>E</BESKZ>
                <SOBSL/>
                <MINBE/>
                <EISBE/>
                <BSTMI/>
                <BSTMA/>
                <BSTFE/>
                <BSTRF/>
                <MABST/>
                <LOSFX/>
                <SBDKZ/>
                <LAGPR/>
                <ALTSL/>
                <KZAUS/>
                <AUSDT/>
                <NFMAT/>
                <KZBED/>
                <MISKZ/>
                <FHORI/>
                <PFREI/>
                <FFREI/>
                <RGEKZ/>
```

```
<FEVOR/>
<BEARZ/>
<RUEZT/>
<TRANZ/>
<BASMG/>
<DZEIT/>
<MAXLZ/>
<LZEIH/>
<KZPRO/>
<GPMKZ/>
<UEETO/>
<UEETK/>
<UNETO/>
<WZEIT/>
<ATPKZ/>
<VZUSL/>
<HERBL/>
<INSMK/>
<SSQSS/>
<KZDKZ/>
<UMLMC/>
<LADGR/>
<XCHPF/>
<USEQU/>
<LGRAD/>
<AUFTL/>
<PLVAR/>
<OTYPE/>
<OBJID/>
<MTVFP/>
<PERIV/>
<KZKFK/>
<VRVEZ/>
<VBAMG/>
<VBEAZ/>
<LIZYK/>
<BWSCL/>
<KAUTB/>
<KORDB/>
<STAWN/>
<HERKL/>
<HERKR/>
<EXPME/>
<MTVER/>
<PRCTR/>
<TRAME/>
<MRPPP/>
<SAUFT/>
<FXHOR/>
<VRMOD/>
<VINT1/>
<VINT2/>
<STLAL/>
<STLAN/>
<PLNNR/>
```

```
                <APLAL/>
                <LOSGR/>
                <SOBSK/>
                <FRTME/>
                <LGPRO/>
                <DISGR/>
                <KAUSF/>
                <QZGTP/>
                <TAKZT/>
                <RWPRO/>
                <COPAM/>
                <ABCIN/>
                <AWSLS/>
                <SERNP/>
                <STDPD/>
                <SFEPR/>
                <XMCNG/>
                <QSSYS/>
                <LFRHY/>
                <RDPRF/>
                <VRBMT/>
                <VRBWK/>
                <VRBDT/>
                <VRBFK/>
                <AUTRU>X</AUTRU>
                <PREFE/>
                <PRENC/>
                <PRENO/>
                <PREND/>
                <PRENE/>
                <PRENG/>
                <ITARK/>
                <PRFRQ/>
                <KZKUP/>
                <STRGR/>
                <LGFSB/>
                <SCHGT/>
                <CCFIX/>
                <EPRIO/>
                <QMATA/>
                <PLNTY/>
                <MMSTA/>
                <SFCPF/>
                <SHFLG/>
                <SHZET/>
                <MDACH/>
                <KZECH/>
                <MMSTD/>
                <MFRGR/>
                <FVIDK/>
                <INDUS/>
                <MOWNR/>
                <MOGRU/>
                <CASNR/>
                <GPNUM/>
```

```
            <STEUC/>
            <FABKZ/>
            <MATGR/>
            <LOGGR/>
            <VSPVB/>
            <DPLFS/>
            <DPLPU/>
            <DPLHO/>
            <MINLS/>
            <MAXLS/>
            <FIXLS/>
            <LTINC/>
            <COMPL/>
            <CONVT/>
            <FPRFM/>
            <SHPRO/>
            <FXPRU/>
            <KZPSP/>
            <OCMPF/>
            <APOKZ/>
            <AHDIS/>
            <EISLO/>
            <NCOST/>
            <MEGRU/>
            <ROTATION_DATE/>
            <UCHKZ/>
            <UCMAT/>
        </E1MARCM>
        <E1MBEWM SEGMENT="1">
            <MSGFN>005</MSGFN>
            <BWKEY>0001</BWKEY>
            <BWTAR/>
            <LVORM/>
            <VPRSV>S</VPRSV>
            <VERPR>0.00</VERPR>
            <STPRS>15.50</STPRS>
            <PEINH>1</PEINH>
            <BKLAS>7920</BKLAS>
            <VMVPR/>
            <VMVER/>
            <VMSTP/>
            <VMPEI/>
            <VMBKL/>
            <VJVPR>S</VJVPR>
            <VJVER>0.00</VJVER>
            <VJSTP>15.50</VJSTP>
            <LFGJA>2002</LFGJA>
            <LFMON>08</LFMON>
            <BWTTY/>
            <ZKPRS/>
            <ZKDAT/>
            <BWPRS/>
            <BWPRH/>
            <VJBWS/>
            <VJBWH/>
```

```
                <VVJLB/>
                <VVMLB/>
                <VVSAL/>
                <ZPLPR/>
                <ZPLP1/>
                <ZPLP2/>
                <ZPLP3/>
                <ZPLD1/>
                <ZPLD2/>
                <ZPLD3/>
                <KALKZ/>
                <KALKL/>
                <XLIFO/>
                <MYPOL/>
                <BWPH1/>
                <BWPS1/>
                <ABWKZ/>
                <PSTAT>BG</PSTAT>
                <KALN1>000100126602</KALN1>
                <KALNR>000100126603</KALNR>
                <BWVA1/>
                <BWVA2/>
                <BWVA3/>
                <VERS1/>
                <VERS2/>
                <VERS3/>
                <HRKFT/>
                <KOSGR/>
                <PPRDZ/>
                <PPRDL/>
                <PPRDV/>
                <PDATZ/>
                <PDATL/>
                <PDATV/>
                <EKALR>X</EKALR>
                <VPLPR>0.00</VPLPR>
                <MLMAA/>
                <MLAST/>
                <VJBKL>7920</VJBKL>
                <VJPEI>1</VJPEI>
                <HKMAT/>
                <EKLAS/>
                <QKLAS/>
                <MTUSE/>
                <MTORG/>
                <OWNPR/>
                <BWPEI>0</BWPEI>
            </E1MBEWM>
        </E1MARAM>
    </IDOC>
</MATMAS02>
</X4Structure>
   </X4IDoc>
</X4SAP>
```

10.9.2.3 SAP Connector (JCo 3)

`SAP Connector (JCo 3)`: Ermöglicht die Übermittlung von Daten vom X4 ESB an ein *SAP R/3*-System. Er erzeugt einen beliebigen Remote Function Call (RFC) bzw. ruft einen beliebigen BAPI-Funktionsbaustein in einem SAP-System via *JCo 3* auf. Zudem können IDocs an einen SAP-Host versandt werden.

Ehemalige Bezeichnung bis Version 4.5.1: *SAP (JCo 3) Request Adapter*

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`Execute`*: Funktions-Adapter ausführen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.sapjco3.SAPAdapter*: Hauptklasse (Standard)
PropertyFile	URL zur Konfigurationsdatei für den Adapter **Mögliche Werte:** Beliebige innerhalb der Java-VM gültige URL (z.B. *`xstore://Project/Folder/Configuration.properties`*)
ReplaceEscapedControlChars	Maskierte Steuerzeichen (z.B.: *`&#31;`*) in den von der JCo-Bibliothek zurückgegebenen XML-Strings durch das Ersetzungszeichen ( ) austauschen **Mögliche Werte:** • `true`: Maskierte Steuerzeichen ersetzen • `false`: Maskierte Steuerzeichen nicht ersetzen

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.9.2.3.1 RFC/BAPI aufrufen

Input

Der Adapter erwartet ein Input-XML-Dokument nach folgendem Muster, das den entsprechenden Funktionsbaustein ggf. mit Inputdaten und den betroffenen Tabellen aufruft:

```
<X4SAP>
 <bapicall name="BAPI-Name" id="IDoc-ID">
   <input>
      <INPUT>
         <!--Parameter und Parameterwerte-->
      </INPUT>
   </input>
   <table>
      (Tabellen)
   </table>
 </bapicall>
</X4SAP>
```

Beispiel

Um BAPI `IDOCTYPE_READ_COMPLETE` in SAP aufzurufen erwartet dieser Adapter folgendes Input-XML-Dokument:

```
<X4SAP>
 <bapicall name="IDOCTYPE_READ_COMPLETE" id="IDoc-ID">
   <input>
      <INPUT>
         <PI_CIMTYP />
         <PI_IDOCTYP> <!--IDoc-Typ--> </PI_IDOCTYP>
         <PI_RELEASE>SY-S</PI_RELEASE>
         <PI_VERSION>3</PI_VERSION>
      </INPUT>
   </input>
   <table>
      <TABLES>
         <PT_FIELDS>
            <SEGMENTTYP />
            <FIELDNAME />
            <INTLEN />
            <EXTLEN />
            <FIELD_POS />
            <BYTE_FIRST />
            <BYTE_LAST />
            <ROLLNAME />
            <DOMNAME />
            <DATATYPE />
            <DESCRP />
            <ISOCODE />
            <VALUETAB />
         </PT_FIELDS>
         <PT_FVALUES>
            <STRNAME />
            <FIELDNAME />
            <FLDVALUE_L />
            <FLDVALUE_H />
            <DESCRP />
         </PT_FVALUES>
         <PT_MESSAGES>
            <MESTYP />
            <DESCRP />
            <IDOCTYP />
            <CIMTYP />
            <RELEASED />
         </PT_MESSAGES>
         <PT_SEGMENTS>
            <NR />
            <SEGMENTTYP />
            <SEGMENTDEF />
            <QUALIFIER />
            <SEGLEN />
            <PARSEG />
            <PARPNO />
            <PARFLG />
            <MUSTFL />
            <OCCMIN />
            <OCCMAX />
            <HLEVEL />
```

```
            <DESCRP />
            <GRP_MUSTFL />
            <GRP_OCCMIN />
            <GRP_OCCMAX />
            <REFSEGTYP />
         </PT_SEGMENTS>
      </TABLES>
   </table>
 </bapicall>
</X4SAP>
```

10.9.2.3.2 IDoc senden

Um ein IDoc an SAP zu senden, ein XML-Dokument wie das folgende als Input für den Adapter verwenden:

```
<X4SAP>
   <X4IDoc type="MATMAS02" description="">
      <X4Mandatory>
         <X4RecipientPartnerType>LS</X4RecipientPartnerType>
         <X4RecipientPartnerNumber>X4_T</X4RecipientPartnerNumber>
         <X4RecipientPartnerFunction />
         <X4SenderPort>SAPT01</X4SenderPort>
         <X4SenderPartnerType>LS</X4SenderPartnerType>
         <X4SenderPartnerNumber>ST01_200</X4SenderPartnerNumber>
         <X4SenderPartnerFunction />
         <X4SenderAddress />
         <X4SenderLogicalAddress />
         <MessageType>
            <MSGTYPE>MATMAS02</MSGTYPE>
         </MessageType>
      </X4Mandatory>
      <X4Structure>
<MATMAS02>
    <IDOC BEGIN="1">
        <EDI_DC40 SEGMENT="1">
            <TABNAM>EDI_DC40</TABNAM>
            <MANDT/>
            <DOCNUM/>
            <DOCREL/>
            <STATUS/>
            <DIRECT/>
            <OUTMOD/>
            <EXPRSS/>
            <TEST/>
            <IDOCTYP>MATMAS02</IDOCTYP>
            <CIMTYP/>
            <MESTYP>MATMAS</MESTYP>
            <MESCOD/>
            <MESFCT/>
            <STD/>
            <STDVRS/>
            <STDMES/>
            <SNDPOR>SAPJCOIDOC</SNDPOR>
            <SNDPRT>LS</SNDPRT>
            <SNDPFC/>
            <SNDPRN>JCOCLNT000</SNDPRN>
            <SNDSAD/>
            <SNDLAD/>
            <RCVPOR/>
            <RCVPRT>LS</RCVPRT>
            <RCVPFC/>
            <RCVPRN>TSTCLNT000</RCVPRN>
            <RCVSAD/>
            <RCVLAD/>
            <CREDAT>20050519</CREDAT>
            <CRETIM>104537</CRETIM>
            <REFINT/>
            <REFGRP/>
            <REFMES/>
```

```
            <ARCKEY/>
            <SERIAL/>
        </EDI_DC40>
        <E1MARAM SEGMENT="1">
            <MSGFN>005</MSGFN>
            <MATNR>BOXCOOKIES</MATNR>
            <ERSDA>20020801</ERSDA>
            <ERNAM>TIGGER</ERNAM>
            <LAEDA/>
            <AENAM/>
            <PSTAT>KBG</PSTAT>
            <LVORM/>
            <MTART>FERT</MTART>
            <MBRSH>L</MBRSH>
            <MATKL>G1113</MATKL>
            <BISMT/>
            <MEINS>PCE</MEINS>
            <BSTME/>
            <ZEINR/>
            <ZEIAR/>
            <ZEIVR/>
            <ZEIFO/>
            <AESZN/>
            <BLATT/>
            <BLANZ>000</BLANZ>
            <FERTH/>
            <FORMT/>
            <GROES/>
            <WRKST/>
            <NORMT/>
            <LABOR/>
            <EKWSL/>
            <BRGEW>0.550</BRGEW>
            <NTGEW>0.000</NTGEW>
            <GEWEI>KGM</GEWEI>
            <VOLUM/>
            <VOLEH/>
            <BEHVO/>
            <RAUBE/>
            <TEMPB/>
            <TRAGR/>
            <STOFF/>
            <SPART/>
            <KUNNR/>
            <WESCH/>
            <BWVOR/>
            <BWSCL/>
            <SAISO/>
            <ETIAR/>
            <ETIFO/>
            <EAN11/>
            <NUMTP/>
            <LAENG/>
            <BREIT/>
            <HOEHE/>
```

```
<MEABM/>
<PRDHA/>
<CADKZ/>
<ERGEW/>
<ERGEI/>
<ERVOL/>
<ERVOE/>
<GEWTO/>
<VOLTO/>
<VABME/>
<KZKFG/>
<XCHPF/>
<VHART/>
<FUELG/>
<STFAK/>
<MAGRV/>
<BEGRU/>
<QMPUR/>
<RBNRM/>
<MHDRZ/>
<MHDHB/>
<MHDLP/>
<VPSTA>KBG</VPSTA>
<EXTWG/>
<MSTAE/>
<MSTAV/>
<MSTDE/>
<MSTDV/>
<KZUMW/>
<KOSCH/>
<NRFHG/>
<MFRPN/>
<MFRNR/>
<BMATN/>
<MPROF/>
<PROFL/>
<IHIVI/>
<ILOOS/>
<KZGVH/>
<XGCHP/>
<COMPL/>
<KZEFF/>
<RDMHD/>
<IPRKZ/>
<PRZUS/>
<MTPOS_MARA/>
<GEWTO_NEW/>
<VOLTO_NEW/>
<WRKST_NEW/>
<AENNR/>
<MATFI/>
<CMREL/>
<SATNR/>
<SLED_BBD/>
<GTIN_VARIANT/>
```

```
            <GENNR/>
            <SERLV/>
            <RMATP/>
            <E1MAKTM SEGMENT="1">
                <MSGFN>005</MSGFN>
                <SPRAS>D</SPRAS>
                <MAKTX>Schachtel mit Keksen</MAKTX>
                <SPRAS_ISO>DE</SPRAS_ISO>
            </E1MAKTM>
            <E1MAKTM SEGMENT="1">
                <MSGFN>005</MSGFN>
                <SPRAS>E</SPRAS>
                <MAKTX>Box of cookies</MAKTX>
                <SPRAS_ISO>EN</SPRAS_ISO>
            </E1MAKTM>
            <E1MARCM SEGMENT="1">
                <MSGFN>005</MSGFN>
                <WERKS>0001</WERKS>
                <PSTAT>BG</PSTAT>
                <LVORM/>
                <BWTTY/>
                <MAABC/>
                <KZKRI/>
                <EKGRP/>
                <AUSME/>
                <DISPR/>
                <DISMM/>
                <DISPO/>
                <PLIFZ>0</PLIFZ>
                <WEBAZ>0</WEBAZ>
                <PERKZ>M</PERKZ>
                <AUSSS>0.00</AUSSS>
                <DISLS/>
                <BESKZ>E</BESKZ>
                <SOBSL/>
                <MINBE/>
                <EISBE/>
                <BSTMI/>
                <BSTMA/>
                <BSTFE/>
                <BSTRF/>
                <MABST/>
                <LOSFX/>
                <SBDKZ/>
                <LAGPR/>
                <ALTSL/>
                <KZAUS/>
                <AUSDT/>
                <NFMAT/>
                <KZBED/>
                <MISKZ/>
                <FHORI/>
                <PFREI/>
                <FFREI/>
                <RGEKZ/>
```

```
<FEVOR/>
<BEARZ/>
<RUEZT/>
<TRANZ/>
<BASMG/>
<DZEIT/>
<MAXLZ/>
<LZEIH/>
<KZPRO/>
<GPMKZ/>
<UEETO/>
<UEETK/>
<UNETO/>
<WZEIT/>
<ATPKZ/>
<VZUSL/>
<HERBL/>
<INSMK/>
<SSQSS/>
<KZDKZ/>
<UMLMC/>
<LADGR/>
<XCHPF/>
<USEQU/>
<LGRAD/>
<AUFTL/>
<PLVAR/>
<OTYPE/>
<OBJID/>
<MTVFP/>
<PERIV/>
<KZKFK/>
<VRVEZ/>
<VBAMG/>
<VBEAZ/>
<LIZYK/>
<BWSCL/>
<KAUTB/>
<KORDB/>
<STAWN/>
<HERKL/>
<HERKR/>
<EXPME/>
<MTVER/>
<PRCTR/>
<TRAME/>
<MRPPP/>
<SAUFT/>
<FXHOR/>
<VRMOD/>
<VINT1/>
<VINT2/>
<STLAL/>
<STLAN/>
<PLNNR/>
```

```
                <APLAL/>
                <LOSGR/>
                <SOBSK/>
                <FRTME/>
                <LGPRO/>
                <DISGR/>
                <KAUSF/>
                <QZGTP/>
                <TAKZT/>
                <RWPRO/>
                <COPAM/>
                <ABCIN/>
                <AWSLS/>
                <SERNP/>
                <STDPD/>
                <SFEPR/>
                <XMCNG/>
                <QSSYS/>
                <LFRHY/>
                <RDPRF/>
                <VRBMT/>
                <VRBWK/>
                <VRBDT/>
                <VRBFK/>
                <AUTRU>X</AUTRU>
                <PREFE/>
                <PRENC/>
                <PRENO/>
                <PREND/>
                <PRENE/>
                <PRENG/>
                <ITARK/>
                <PRFRQ/>
                <KZKUP/>
                <STRGR/>
                <LGFSB/>
                <SCHGT/>
                <CCFIX/>
                <EPRIO/>
                <QMATA/>
                <PLNTY/>
                <MMSTA/>
                <SFCPF/>
                <SHFLG/>
                <SHZET/>
                <MDACH/>
                <KZECH/>
                <MMSTD/>
                <MFRGR/>
                <FVIDK/>
                <INDUS/>
                <MOWNR/>
                <MOGRU/>
                <CASNR/>
                <GPNUM/>
```

```
            <STEUC/>
            <FABKZ/>
            <MATGR/>
            <LOGGR/>
            <VSPVB/>
            <DPLFS/>
            <DPLPU/>
            <DPLHO/>
            <MINLS/>
            <MAXLS/>
            <FIXLS/>
            <LTINC/>
            <COMPL/>
            <CONVT/>
            <FPRFM/>
            <SHPRO/>
            <FXPRU/>
            <KZPSP/>
            <OCMPF/>
            <APOKZ/>
            <AHDIS/>
            <EISLO/>
            <NCOST/>
            <MEGRU/>
            <ROTATION_DATE/>
            <UCHKZ/>
            <UCMAT/>
        </E1MARCM>
        <E1MBEWM SEGMENT="1">
            <MSGFN>005</MSGFN>
            <BWKEY>0001</BWKEY>
            <BWTAR/>
            <LVORM/>
            <VPRSV>S</VPRSV>
            <VERPR>0.00</VERPR>
            <STPRS>15.50</STPRS>
            <PEINH>1</PEINH>
            <BKLAS>7920</BKLAS>
            <VMVPR/>
            <VMVER/>
            <VMSTP/>
            <VMPEI/>
            <VMBKL/>
            <VJVPR>S</VJVPR>
            <VJVER>0.00</VJVER>
            <VJSTP>15.50</VJSTP>
            <LFGJA>2002</LFGJA>
            <LFMON>08</LFMON>
            <BWTTY/>
            <ZKPRS/>
            <ZKDAT/>
            <BWPRS/>
            <BWPRH/>
            <VJBWS/>
            <VJBWH/>
```

```
                <VVJLB/>
                <VVMLB/>
                <VVSAL/>
                <ZPLPR/>
                <ZPLP1/>
                <ZPLP2/>
                <ZPLP3/>
                <ZPLD1/>
                <ZPLD2/>
                <ZPLD3/>
                <KALKZ/>
                <KALKL/>
                <XLIFO/>
                <MYPOL/>
                <BWPH1/>
                <BWPS1/>
                <ABWKZ/>
                <PSTAT>BG</PSTAT>
                <KALN1>000100126602</KALN1>
                <KALNR>000100126603</KALNR>
                <BWVA1/>
                <BWVA2/>
                <BWVA3/>
                <VERS1/>
                <VERS2/>
                <VERS3/>
                <HRKFT/>
                <KOSGR/>
                <PPRDZ/>
                <PPRDL/>
                <PPRDV/>
                <PDATZ/>
                <PDATL/>
                <PDATV/>
                <EKALR>X</EKALR>
                <VPLPR>0.00</VPLPR>
                <MLMAA/>
                <MLAST/>
                <VJBKL>7920</VJBKL>
                <VJPEI>1</VJPEI>
                <HKMAT/>
                <EKLAS/>
                <QKLAS/>
                <MTUSE/>
                <MTORG/>
                <OWNPR/>
                <BWPEI>0</BWPEI>
            </E1MBEWM>
        </E1MARAM>
    </IDOC>
</MATMAS02>
</X4Structure>
   </X4IDoc>
</X4SAP>
```

10.9.2.4 SAP Explorer (JCo 2)

`SAP Explorer (JCo 2)`: Erzeugt Strukturbeschreibungen von IDocs und BAPIs, listet BAPIs oder IDocs oder erzeugt leere IDoc- und BAPI-Strukturen und gibt je nach Operation ein entsprechendes XML-Dokument aus.

Ehemalige Bezeichnung bis Version 4.5.1: *SAP (JCO 2) Explorer Adapter*

> **Bitte beachten Sie:**
> Zur Nutzung des *SAP Explorers (JCo 2)* werden Rechte zur Remote-Ausführung folgender RFC-Bausteine benötigt:
>
> - `RPY_BOR_TREE_INIT`
> - `IDOCTYPES_LIST_WITH_MESSAGES`
> - `IDOCTYPE_READ_COMPLETE`
> - `RPY_OBJECTTYPE_READ`
> - `RFC_FUNCTION_SEARCH`

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`Get BapiTree`*: Liste aller verfügbaren BAPIs ausgeben • *`Get IDocList`*: Liste aller verfügbaren IDocs ausgeben • *`Read Bapi`*: Beschreibung der BAPI-Struktur ausgeben • *`Read IDoc`*: Beschreibung der IDoc-Struktur ausgeben • *`Create Empty Bapi`*: Leere BAPI-Struktur zur Verwendung in der X4 BPMS erzeugen • *`Create Empty Idoc`*: Leere IDoc-Struktur zur Verwendung in der X4 BPMS erzeugen • *`SearchForBapis`*: Liste aller BAPIs ausgegeben, die dem Suchausdruck entsprechen • *`Nothing`*: Input unverändert ausgeben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.sapadapter.SAPBapiIdocNotationExplorer*: Hauptklasse (Standard)
PropertyFile	URL zur Konfigurationsdatei für den Adapter (hebt die Adapter-Konfiguration auf) **Mögliche Werte:** Beliebige innerhalb der Java-VM gültige URL (z. B. *xstore://Project/Folder/Configuration.properties*)
Client	SAP-Mandant
User	Benutzername
Password	Passwort
Language	Anmeldesprache
SystemNumber	SAP-System-Nummer
RouterString	Router-String
SystemType	Art des entfernten Hosts **Mögliche Werte:** • *2*: R/2 • *3*: R/3 • *E*: Extern

Statuswerte

`-1`	Die Operation wurde erfolgreich ausgeführt.
`1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Funktions-Adapter erwartet je nach gewählter Operation z.T. eine adapterspezifische Input-XML-Struktur.

Operation	
Get BapiTree	Für die Operation `Get BapiTree` wird kein Input-XML-Dokument benötigt
Get IDocList	Für die Operation `Get IDocList` wird kein Input-XML-Dokument benötigt

Operation	
Read Bapi	Für die Operation `Read Bapi` erwartet dieser Funktions-Adapter folgendes Input-XML-Dokument: `<Object><!--BAPI-ID--></Object>`
Read IDoc	Für die Operation `Read IDoc` erwartet dieser Funktions-Adapter folgendes Input-XML-Dokument: `<ReadIDoc><!--IDoc-ID--></ReadIDoc>`
Create Empty Bapi	Für die Operation `Create Empty Bapi` erwartet dieser Funktions-Adapter folgendes Input-XML-Dokument: `<Bapiname><!--Name des BAPIs--></Bapiname>`
Create Empty Idoc	Für die Operation `Create Empty Idoc` erwartet dieser Funktions-Adapter folgendes Input-XML-Dokument: `<IDoc><!--Name des IDocs--></IDoc>`
SearchForBapis	Für die Operation `SearchForBapis` erwartet dieser Funktions-Adapter folgendes Input-XML-Dokument: `<Search functionname="BAPI-/Funktionsname" />`
Nothing	Für die Operation `Nothing` wird kein Input-XML-Dokument benötigt.

10.9.2.5 SAP Explorer (JCo 3)

`SAP Explorer (JCo3):` Erzeugt Strukturbeschreibungen von IDocs und BAPIs, listet BAPIs oder IDocs oder erzeugt leere IDoc- und BAPI-Strukturen und gibt je nach Operation ein entsprechendes XML-Dokument aus.

Ehemalige Bezeichnung bis Version 4.5.1: *SAP (JCo 3) Explorer Adapter*

Bitte beachten Sie:
Zur Nutzung des *SAP Explorers (JCo 3)*werden Rechte zur Remote-Ausführung folgender RFC-Bausteine benötigt:

- `RPY_BOR_TREE_INIT`
- `IDOCTYPES_LIST_WITH_MESSAGES`
- `IDOCTYPE_READ_COMPLETE`
- `RPY_OBJECTTYPE_READ`
- `RFC_FUNCTION_SEARCH`

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`GetBapiTree`*: Liste aller verfügbaren BAPIs ausgeben • *`GetIDocList`*: Liste aller verfügbaren IDocs ausgeben • *`ReadBapi`*: Beschreibung der BAPI-Struktur ausgeben • *`ReadIDoc`*: Beschreibung der IDoc-Struktur ausgeben • *`CreateEmptyBapi`*: Leere BAPI-Struktur zur Verwendung in der X4 BPMS erzeugen • *`CreateEmptyIdoc`*: Leere IDoc-Struktur zur Verwendung in der X4 BPMS erzeugen • *`SearchForBapis`*: Liste aller BAPIs ausgegeben, die dem Suchausdruck entsprechen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.sapjco3.SAPBapiIdocNotationExplorer*: Hauptklasse (Standard)
PropertyFile	URL zur Konfigurationsdatei für den Adapter **Mögliche Werte:** Beliebige innerhalb der Java-VM gültige URL (z.B. *`xstore://Project/Folder/Configuration.properties`*)
Client	SAP-Mandant (jco.client.client in der Properties-Datei)
User	Benutzername (jco.client.user in der Properties-Datei)

Password	Passwort (jco.client.passwd in der Properties-Datei)
Language	Anmeldesprache (jco.client.lang in der Properties-Datei)
SystemNumber	SAP-System-Nummer (jco.client.sysnr in der Properties-Datei)
RouterString	Router-String (jco.client.ashost in der Properties-Datei)
SystemType	Art des entfernten Hosts (`jco.client.type` in der Properties-Datei) **Mögliche Werte:** • *2*: R/2 • *3*: R/3 • *E*: Extern
ReplaceEscapedControlChars	Maskierte Steuerzeichen (z.B.: *&#31;*) in den von der JCo-Bibliothek zurückgegebenen XML-Strings durch das Ersetzungszeichen ( ) austauschen **Mögliche Werte:** • `true`: Maskierte Steuerzeichen ersetzen • `false`: Maskierte Steuerzeichen nicht ersetzen

Statuswerte

`-1`	Die Operation wurde erfolgreich ausgeführt.
`1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Funktions-Adapter erwartet je nach gewählter Operation z.T. eine adapterspezifische Input-XML-Struktur.

Operation	
Get BapiTree	Für die Operation `Get BapiTree` wird kein Input-XML-Dokument benötigt.
Get IDocList	Für die Operation `Get IDocList` wird kein Input-XML-Dokument benötigt.
Read Bapi	Für die Operation `Read Bapi` erwartet dieser Funktions-Adapter folgendes Input-XML-Dokument: `<Object><!--BAPI-ID--></Object>`

Operation	
Read IDoc	Für die Operation `Read IDoc` erwartet dieser Funktions-Adapter folgendes Input-XML-Dokument: `<ReadIDoc><!--IDoc-ID--></ReadIDoc>`
Create Empty Bapi	Für die Operation `Create Empty Bapi` erwartet dieser Funktions-Adapter folgendes Input-XML-Dokument: `<Bapiname><!--Name des BAPIs--></Bapiname>`
Create Empty Idoc	Für die Operation `Create Empty Idoc` erwartet dieser Funktions-Adapter folgendes Input-XML-Dokument: `<IDoc><!--Name des IDocs--></IDoc>`
SearchForBapis	Für die Operation `SearchForBapis` erwartet dieser Funktions-Adapter folgendes Input-XML-Dokument: `<Search functionname="BAPI-/Funktionsname" />`

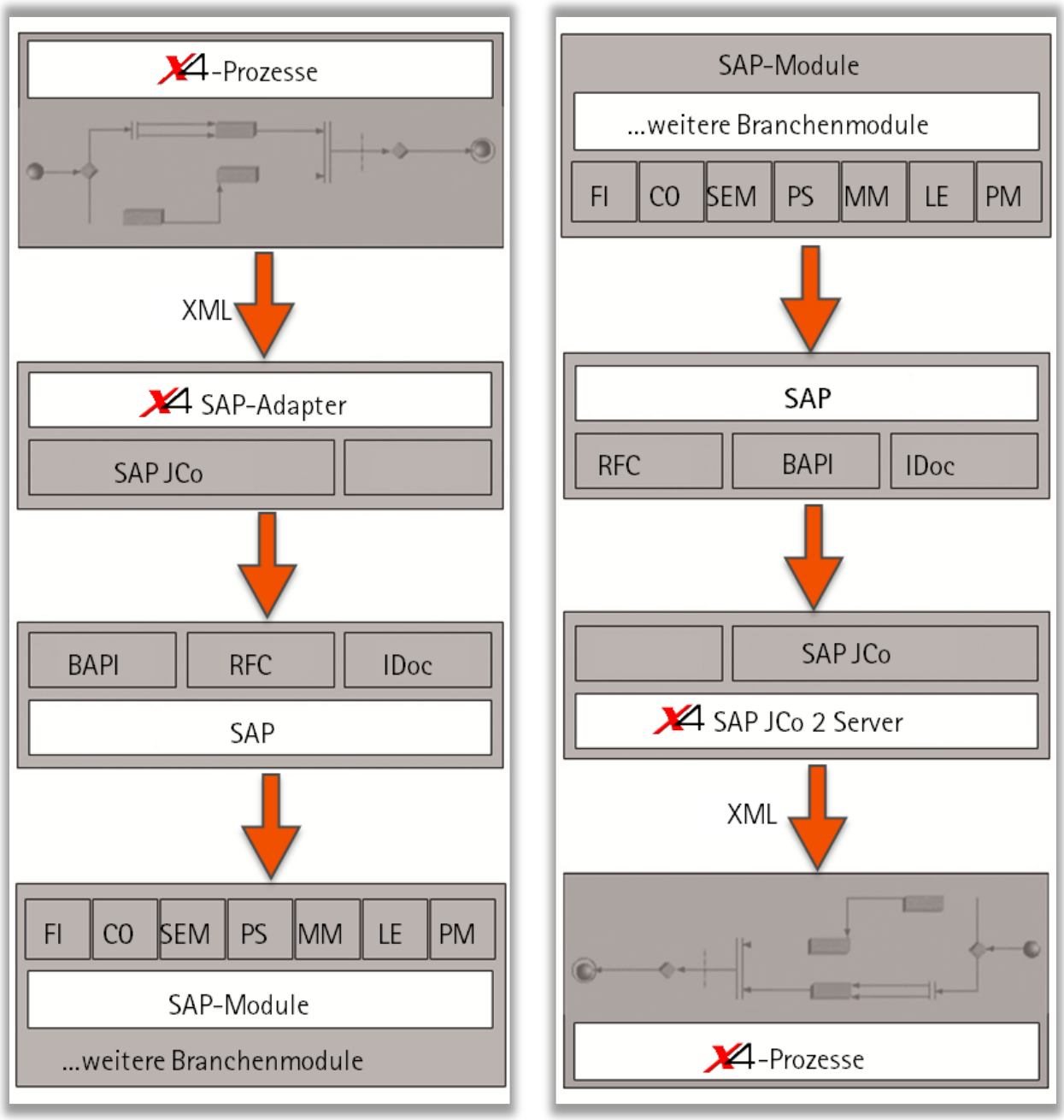
10.9.2.6 SAP Listener (JCo 2)

Der *SAP Listener (JCo)* ermöglicht die Verarbeitung von eingehenden Kommunikationsanfragen via BAPI/RFC und von IDocs eines SAP-Systems. Er stellt dabei automatisch ggf. unterbrochene Verbindungen wieder her.
Ehemalige Bezeichnung bis Version 4.5.1: *X4 SAP JCo 2 Listener*

Funktionen von X4 SAP JCo 2

Der *X4 Server* ermöglicht die bidirektionale Anbindung von *SAP R/3*-Systemen über den *SAP Java Connector* in Version 2 (*JCo 2*). Dabei sind folgende Komponenten relevant:

- *SAP JCo 2 Server*: Ermöglicht die Verarbeitung von eingehenden Kommunikationsanfragen via BAPI/RFC und von IDocs eines SAP-Systems an den *X4 Server*. Der *SAP JCo 2 Server* stellt dabei automatisch ggf. unterbrochene Verbindungen wieder her.
- *SAP*-Adapter: In *X4*-Prozessen kann dieser Funktions-Adapter einen beliebigen Remote Function Call (RFC) bzw. einen beliebigen BAPI-Funktionsbaustein in einem SAP-System aufrufen. Zudem können IDocs an *SAP R/3* versandt werden, siehe *Adapter und Bausteine, SAP Connector (JCo 2)*.
- *SAP Explorer (JCo2)*: Dieser ermöglicht während der Prozess-Entwicklung das Erzeugen von Strukturbeschreibungen von IDocs und BAPIs. Zudem listet er BAPIs oder IDocs oder erzeugt leere IDoc- und BAPI-Strukturen und gibt je nach Operation ein entsprechendes XML-Dokument aus, siehe *Adapter und Bausteine, SAP Explorer (JCo 2)*.



SAP Java Connector

Über den *JCo 2 Server* kann mit der X4 BPMS ein SAP-System angebunden werden. Es können auch mehrere SAP-Systeme über eine jeweils separate *JCo Server*-Instanz angebunden werden. Zum Verwenden des *JCo 2*-Servers ist eine entsprechende Konfiguration erforderlich:

- *X4*-seitige Konfiguration in `X4config.xml`, siehe SAP JCo 2 Server-Konfiguration
- Definition von Verbindungs-Eigenschaften, siehe Properties für den JCo 2 Server
- Zuordnung der Funktionsbaustein-Aufrufe zu Technical Processes, siehe Funktionsbaustein-/Methodendefinitionen (X4 SAP JCo 2)

Identifikation von JCo 2-Servern über ProgID

Jede *JCo 2 Server*-Instanz bzw. jedes an SAP angebundene System kommuniziert über eine eindeutige *ProgID*, die in der Server-Properties-Datei, standardmäßig `jco_server_settings.prop`, konfiguriert wird und beim Start der Server-Instanz dem SAP-System bekannt gemacht wird. Pro SAP-System können mehrere ProgIDs gleichzeitig verwendet werden. Über die *ProgID* werden IDocs und RFC/BAPIs an das jeweilige Ziel-System eindeutig "adressiert".

10.9.2.6.1 Installation von X4 SAP JCo 2

Welche Schritte zur Installation von *X4 SAP JCo 2* erforderlich sind

> **Beispiel**
> Als Beispiel wird eine Installation für den *JBoss* Applikations-Server auf Basis eines 32-Bit-Windows-Systems beschrieben.

10.9.2.6.1.1 Voraussetzungen

Zur Installation eines *SAP JCo 2 Server* im *X4 Server* werden *SAP*-Bibliotheken benötigt, die z. T. betriebssystem- und architekturabhängig sind. Diese Bibliotheken werden auch für den SAP Connector (JCo 2) und den SAP Explorer (JCo 2) benötigt, die bereits in `x4.ear` enthalten sind.

Als registrierter SAP-Partner können Sie die erforderlichen Bibliotheken sowie weiterführende Dokumentation über das *SAPNet* von http://service.sap.com/connectors herunterladen.

> Wählen Sie hierbei zueinander kompatible Versionsnummern der Bibliotheken (diese sind zudem betriebssystem- und architekturabhängig).
> Achten Sie darauf, dass das verwendete Betriebsystem kompatibel mit der eingesetzten JDK-Version und der Version Ihres SAP-Systems ist. Folgende Matrix zeigt den Zusammenhang dieser drei Variablen.

		OS		**SAP**	
		32 bit	64 bit	32 bit	64 bit
JDK	32 bit	x	x	x	-
	64 bit	-	x	-	x

10.9.2.6.1.2 Bibliotheken installieren

1. Native Bibliotheken (je nach System) in den entsprechenden Ordner des Applikations-Servers für native Binär-Abhängigkeiten oder in das `bin`-Verzeichnis von Java kopieren.
   Beispiel: Unter Windows 32-Bit: `librfc32.dll`, `msvcr71.dll`, `msvcp71.dll` und `sapjcorfc.dll` nach `C:\Windows\System32` oder nach `C:\X4\Server\jdk1.7\bin`) kopieren

2. Folgende Java-Bibliotheken in den Classpath des Applikations-Servers kopieren:
   - `sapidoc.jar`
   - `sapjco.jar`
   - `sapidocjco.jar`

   Beispiel JBoss 4.2.3.GA: Diese Bibliotheken nach `C:\X4\JBoss4.2.3.GA\server\default\lib` kopieren.
   Beispiel JBoss AS 7.1.1: Diese Bibliotheken unter `C:\X4\Server\jboss-as-7.1.1.Final\standalone\deployments` in das `lib`-Verzeichnis von `X4.ear` kopieren.
3. *X4 Server* neu starten.

10.9.2.6.1.3 Installation prüfen

Über den *SAP Explorer (JCo 2)* (siehe *X4 Experience*-Beispiel `X4Experience/AdapterSuites/X4SAPSuite/ SAP(JCo2)Explorer(Prototype)/Processes/CreateEmptyBapi.wrf`) können Sie prüfen, ob eine Verbindung aufgebaut werden kann:

1. Sicherstellen, dass die Konfiguration des *SAP JCo 2 Server* vollständig ist, siehe *X4 Server-Konfiguration (X4config.xml)*.
2. Adapterparameter `SystemType` auf *3* setzen (für eine Verbindung zu *SAP R/3*).
3. Sicherstellen, dass Adapteroperation `Create Empty Bapi` gewählt ist.
4. Prozess mit dem Input-XML `<Bapiname>BAPI's name</Bapiname>` ausführen und Ergebnis-XML prüfen.

10.9.2.6.2 Konfiguration des X4 SAP Listeners (JCo 2)

Welche Konfiguration *X4*-seitig angepasst werden muss, um den *JCo 2 Server* zu starten, von außen erreichbar zu machen und um den eingehenden RFC/BAPIs und IDocs entsprechende Handler-Prozesse in der X4 BPMS zuzuordnen, erfahren Sie im Folgenden.

10.9.2.6.2.1 SAP JCo 2 Server-Konfiguration

Im Element `<jcoServer>` können Sie in `X4config.xml` die *SAP Java Connector*-Einstellungen für den *X4 Server* konfigurieren.

> Ab der *X4 Server*-Revision *27282* besitzt das Element `<jcoServer>` eine veränderte Struktur: IDoc-Handler und Function-Handler wurden in die *JCo 2 Server Properties* ausgelagert, siehe Properties für den JCo 2 Server. Zudem ist nun möglich, *mehrere* Properties-Dateien und Funktionsbaustein-/Methoden-Definitionen in `X4config.xml` anzugeben, um Verbindungen zu mehreren SAP-Systemen zu ermöglichen.
> Falls mehrere Properties-Dateien und Funktionsbaustein-/Methoden-Definitionen angegeben werden, erfolgt deren Zuordnung über die Reihenfolge. d. h. die erste *JCo Server*-Properties-Datei in `<propertiesURL>` wird der ersten Funktionsbaustein-/Methoden-Definition in `<methodsURL>` zugewiesen, die zweite *JCo Server*-Properties-Datei wird der zweiten Funktionsbaustein-/Methoden-Definition zugewiesen etc.

Element <jcoServer>

`<propertiesURL>`	URL zu einer Properties-Datei, die *JCo 2 Server*-Einstellungen enthält; Ab *X4 Server*-Version 4.2 können Sie hier mehrere durch Semikolon getrennte URLs zu verschiedenen Properties-Dateien angeben **Mögliche Werte**: Beliebige gültige URL (z. B. *`file:///C:/X4/X4DB/0/jco_server_settings.prop`*)
`<methodsURL>`	URL zu einem XML-Dokument, das die Zuordnung von Funktionsbausteinen/Methoden und dem entsprechenden Verarbeitungsprozess im *X4 Server* enthält; Ab *X4 Server*-Version 4.2 können Sie hier mehrere durch Kommas getrennte URLs zu verschiedenen Konfigurations-Dokumenten angeben **Mögliche Werte**: Beliebige gültige URL (z. B. *`file:///C:/X4/X4DB/0/jco_method_def.xml`*)

Element <sapadapter>
Im Element `<sapadapter>`, das dem Element `<jcoServer>` in `X4config.xml` nachfolgt, können Sie die Standard-Verbindungseinstellungen für den *SAP Connector (JCo 2)* konfigurieren. Diese wird als Fallback-Konfiguration verwendet.

`<propertiesURL>`	URL zu einer Properties-Datei, die Verbindungseinstellungen enthält, die herangezogen werden, wenn der *SAP Connector (JCo 2)*keine Einstellungen findet.

10.9.2.6.2.2 Properties für den JCo 2 Server

In einem Konfigurations-Dokument (standardmäßig `<X4>/X4DB/0/jco_server_settings.prop`) werden die Anzahl der *JCo Server*-Instanzen, Funktions-Handler-Prozesse und IDoc-Handler-Prozesse definiert. In der zentralen Konfigurationsdatei `<X4>/X4config.xml` muss für jede *JCo Server*-Instanz jeweils eine *JCo Server*-Konfigurationsdatei referenziert werden. Die Properties-Datei ist in folgende Blöcke unterteilt.

Allgemeine Konfiguration
Mit den folgenden Properties lässt sich der *JCo 2 Server* hinsichtlich der zulässigen Server-Instanzen und dem Umgang mit Verbindungen und Connection Pooling konfigurieren, siehe Connection Pooling (X4 SAP JCo 2).

```
# Number of server instances
server_instances_count = Anzahl der JCo Server-Instanzen
usePool                = Boolescher Wert
poolSize               = Zulässige Zahl der Verbindungen
```

Erläuterung:

- `server_instances_count`: Anzahl der *JCo Server*-Instanzen
- `usePool`: Connection Pooling aktivieren (*true*) oder deaktivieren (*false*), siehe Connection Pooling (X4 SAP JCo 2)
- `poolSize`: Anzahl der zulässigen Client-Verbindungen im Pool; Empfehlung: Einen Wert wählen, der mindestens der Summer der *JCo 2 Server*-Instanzen (identifiziert durch je eine *ProgID*) entspricht.

Funktions-Handler-Konfiguration
In den folgenden Properties werden die Funktions-Handler für RFC/BAPI-Aufrufe pro SAP-Host definiert. Die Konfiguration der Standard-Funktions-Handler-Prozesse ist optional und dient lediglich als Fallback, falls in den Funktionsbaustein-/Methodendefinitionen keine entsprechenden Handler-Prozesse zugeordnet wurden, siehe Funktionsbaustein-/Methodendefinitionen (X4 SAP JCo 2).

Commit- und Rollback-Handler können jedoch nur über diese Properties vereinbart werden.

```
# per jcoserver instance settings
# this should always be set in the method definitions
jcoserver.function.handler.userId =   Benutzerordner in X4DB (optional)
jcoserver.function.handler.process =  Standard-Function-Handler-Prozess (optional)

jcoserver.function.commit.userId =    Benutzer-Repository für Commit-Prozess
jcoserver.function.commit.process =   Pfad zum X4-Prozess für Commit
jcoserver.function.rollback.userId =  Benutzer-Repository für Rollback-Prozess
jcoserver.function.rollback.process = Pfad zum X4-Prozess für Rollback
```

Erläuterung:

- `jcoserver.function.handler.userId`: Benutzer-Repository in X4DB (z. B. *1*), in dem der Funktionsaufruf-Handler-Prozess liegt
- `jcoserver.function.handler.process`: optionale Pfadangabe zu einem *X4*-Prozess, der als Standard-Handler für Funktionsaufrufe dient
- `jcoserver.function.commit.userId`: Benutzer-Repository in X4DB (z. B. *1*), in dem der *Commit*-Handler-Prozess für Funktionsaufrufe liegt
- `jcoserver.function.commit.process`: Pfadangabe zu einem *X4*-Prozess, der als *Commit*-Handler für Funktionsaufrufe dient
- `jcoserver.function.rollback.userId`: Benutzer-Repository in X4DB (z. B. *1*), in dem der *Rollback*-Handler-Prozess für Funktionsaufrufe liegt
- `jcoserver.function.rollback.process`: Pfadangabe zu einem *X4*-Prozess, der als *Rollback*-Handler für Funktionsaufrufe dient

IDoc-Verarbeitung
In den folgenden Properties werden die Handler für eingehende IDocs definiert. Die Konfiguration des Standard-IDoc-Handler-Prozesses ist optional und dient lediglich als Fallback, falls in den Funktionsbaustein-Definitionen keine entsprechenden Handler-Prozesse zugeordnet wurden, siehe Funktionsbaustein-/Methodendefinitionen (X4 SAP JCo 2).

Pro *JCo 2 Server*-Instanz (identifiziert durch je eine *ProgID*) kann jeweils ein Commit- und Rollback-Handler über diese Properties vereinbart werden.

> ⓘ Wenn keine Handler-Konfiguration gefunden wird, dann wird der entsprechende Commit bzw. Rollback übersprungen.

```
# this should always be set in the method definitions
jcoserver.idoc.handler.userId   = Benutzerordner in X4DB (optional)
jcoserver.idoc.handler.process  = Standard-IDoc-Handler-Prozess (optional)

# commit and rollback handlers for IDocs
jcoserver.idoc.commit.userId  =   Benutzer-Repository für Commit-Prozess
jcoserver.idoc.commit.process =   Pfad zum X4-Prozess für Commit
jcoserver.idoc.rollback.userId  = Benutzer-Repository für Rollback-Prozess
jcoserver.idoc.rollback.process = Pfad zum X4-Prozess für Rollback
```

Erläuterung:

- `jcoserver.idoc.handler.userId`: Optionale Angabe des Benutzer-Repositorys in X4DB (z. B. *1*), in dem der IDoc-Handler-Prozess liegt (Fallback)
- `jcoserver.idoc.handler.process`: Optionale Pfadangabe zu einem *X4*-Prozess, der als Standard-Handler für IDocs dient (Fallback)
- `jcoserver.idoc.commit.userId`: Benutzer-Repository in X4DB (z. B. *1*), in dem der *Commit*-Handler-Prozess für IDocs liegt
- `jcoserver.idoc.commit.process`: Pfadangabe zu einem *X4*-Prozess, der als *Commit*-Handler für IDocs dient
- `jcoserver.idoc.rollback.userId`: Benutzer-Repository in X4DB (z. B. *1*), in dem der *Rollback*-Handler-Prozess für IDocs liegt
- `jcoserver.idoc.rollback.process`: Pfadangabe zu einem *X4*-Prozess, der als *Rollback*-Handler für IDocs dient

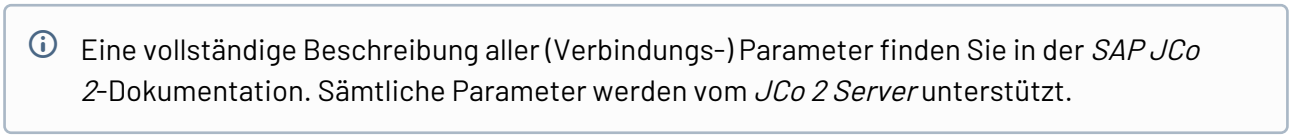
Verbindungs-Einstellungen zum SAP-Host
In den folgenden Properties werden die Einstellungen für den Verbindungsaufbau zu einem SAP-Host für die jeweilige *JCo 2 Server*-Instanz definiert:

```
# connection String to SAP
jco.server.gwhost  = Gateway-Host
jco.server.gwserv  = Gateway-Service
jco.server.progid  = Programm-ID
jco.server.trace   = Boolescher Wert
jco.server.unicode = Boolescher Wert

# required for dynamic repository and metadata lookup
jco.client.client  = Client-ID
jco.client.user    = Benutzername
jco.client.passwd  = Passwort
jco.client.lang    = Client-Sprache
jco.client.sysnr   = SAP-Systemnummer
jco.client.ashost  = SAP-Applikations-Server
jco.client.trace   = Boolescher Wert
```

Erläuterung:

- `jco.server.gwhost`: Gateway-Host
- `jco.server.gwserv`: Gateway-Service (normalerweise *sapgwNN*)
- `jco.server.progid`: Programm-ID des SAP-Hosts
- `jco.server.trace`: RFC Trace für den SAP-Server aktivieren (*1*) oder deaktivieren (*0*)
- `jco.server.unicode`: Im Unicode-Modus verbinden (*1*) oder nicht (*0*)
- `jco.client.client`: Client-ID (dreistelliger numerischer Wert)
- `jco.client.user`: Benutzername
- `jco.client.passwd`: Passwort
- `jco.client.lang`: Sprache des Clients
- `jco.client.sysnr`: SAP-Systemnummer
- `jco.client.ashost`: SAP-Applikations-Server
- `jco.client.trace`: RFC Trace aktivieren (*1*) oder deaktivieren (*0*)



Eine vollständige Beschreibung aller (Verbindungs-) Parameter finden Sie in der *SAP JCo 2*-Dokumentation. Sämtliche Parameter werden vom *JCo 2 Server* unterstützt.

10.9.2.6.2.3 Funktionsbaustein-/Methodendefinitionen (X4 SAP JCo 2)

In einem Konfigurations-Dokument (standardmäßig `<X4>/X4DB/0/jco_method_def.xml`) kann für jeden Funktionsbaustein (Methode), die vom SAP-Host angesprochen wird, über den *JCo 2 Server* ein entsprechender *X4*-Prozess zur Weiterverarbeitung zugeordnet werden.
In der zentralen Konfigurationsdatei `<X4>/X4config.xml` muss für jede *JCo Server*-Instanz jeweils eine Funktionsbaustein-/Methodendefinition referenziert werden.

Erwartete XML-Struktur
In der Funktionsbaustein-/Methodendefinition können Sie innerhalb des Wurzel-Elements `<definitions>` beliebig viele Elemente `<method>` nach folgendem Muster anlegen:

```
<method>
  <name><!--Funktionsbaustein-/Methodenname--></name>
    <wait_for_result><!--Boolescher Wert--></wait_for_result>
    <class>de.softproject.integration.sapjcoserver.requesthandlers.WorkflowStarter</
class>
    <properties>
      <property Name="workflowname"><!--Pfad zum X4-Prozess--></property>
      <property Name="userID"><!--Benutzerordner in X4DB--></property>
      <property Name="raw"><!--Boolescher Wert--></property>
    </properties>
</method>
```

Für jede Funktionsbaustein-/Methodendefinition gilt:

- Element `<name>` enthält den Namen des durch SAP aufgerufenen Funktionsbaustein (Methode); Der Platzhalter `*` steht für alle anderen Funktionsbausteine , für die kein expliziter Funktionsbaustein definiert wurde.
- In Element `<wait_for_result>` können Sie konfigurieren, ob SAP auf die Antwort warten soll (*false*), oder ob der Funktionsbaustein-/Methodenaufruf asynchron erfolgt (*true*).
- Element `<class>` gibt die verwendete Klasse an, die zur Verarbeitung des Funktionsbausteins (bzw. der Methode) hinterlegt ist (normalerweise ist dies `de.softproject.integration.sapjcoserver.requesthandlers.WorkflowStarter` zum Starten von *X4*-Prozessen); in den Kindelementen von `<class>` geben Sie die zugehörigen Eigenschaften in jeweils einem Element `<property>` an.

Für die *X4*-Prozess-Starterklasse werden folgende Eigenschaften angegeben:

- `workflowname`: Pfad zum *X4*-Prozess, der die Verarbeitung des Funktionsbaustein-/Methodenaufrufs übernimmt
- `userID`: Repository-Benutzerordner, in dem der *X4*-Prozess liegt
- `raw`: IDoc im Raw-Format verarbeiten (*true*) oder (*false*)

> ⓘ In älteren Funktionsbaustein-/Methodendefinitionen finden sich ggf. zusätzliche Angaben in einem Element `<type>` und in einem Element `<property Name="languageID">`. Diese besitzen keine Funktion und können daher weggelassen werden.

Beispiel
Im folgenden Beispiel wird für den Funktionsbaustein (Methode) ORDERS05 der *X4*-Prozess save_idoc.wrf aufgerufen und es werden Daten im *IDoc Raw*-Format für die nachgeordnete Verarbeitung bereitgestellt.

```
<method>
  <name>ORDERS05</name>
    <wait_for_result>false</wait_for_result>
    <class>de.softproject.integration.sapjcoserver.requesthandlers.
           WorkflowStarter</class>
    <properties>
      <property Name="workflowname">Project/Folder/ProcessIDoc.wrf</property>
      <property Name="userID">1</property>
      <property Name="raw">true</property>
    </properties>
</method>
```

10.9.2.6.2.4 Connection Pooling (X4 SAP JCo 2)

Der *SAP JCo 2 Server* kann für jede Instanz einen Pool für Client-Verbindungen bereitstellen. Wenn kein Pool existiert, dann wird direkt eine *JCo Client*-Verbindung hergestellt.
Einstellungen zum Connection Pooling in der Server-Properties-Konfigurationsdatei, siehe Properties für den JCo 2 Server:

- `usePool`: Connection Pooling aktivieren (*`true`*) oder deaktivieren (*`false`*)
- `poolSize`: Anzahl der zulässigen Client-Verbindungen im Pool; Empfehlung: Einen Wert wählen, der mindestens der Summer der *JCo 2 Server*-Instanzen (identifiziert durch je eine *ProgID*) entspricht.

> **Poolgröße und Anzahl SAP-Adapter**
> Erhöhen Sie den in `poolSize` angegebenen Wert um die Anzahl der in *X4*-Prozessen eingesetzten SAP-Adapter. Im SAP Connector (JCo 2) können Sie den zu verwendenden Pool über den Adapter-Parameter `x4.connectionPoolUID` (frei wählbar) festlegen.

10.9.2.6.2.5 JCo 2 Server-Konfiguration aktualisieren

Um geänderte Konfigurationen am *SAP JCo Server* zu aktualisieren, ist es nicht erforderlich, den kompletten *X4 Server* neuzustarten. Der *X4 Server* stellt hierzu eine JMX-MBean `de.softproject.X4.X4Management` mit folgenden drei Methoden bereit.

> Bei *JBoss 4.2.3 GA* können Sie die JMX-Konsole über eine URL nach dem Muster `http://`*`host`*`:`*`port`*`/jmx-console/` im Browser aufrufen, z. B. http://localhost:8080/jmx-console/. Dort finden Sie die entsprechende MBean `de.softproject.X4.X4Management`.

JCo Server neu starten:	Methode `restartSAPJcoServer()` über **Invoke** aufrufen.
JCo Server herunterfahren:	Methode `stopSAPJcoServer()` über **Invoke** aufrufen.
JCo Server-Konfiguration auf Änderungen prüfen und diese einlesen:	Methode `reloadChangedSAPJCoServers()` über **Invoke** aufrufen.

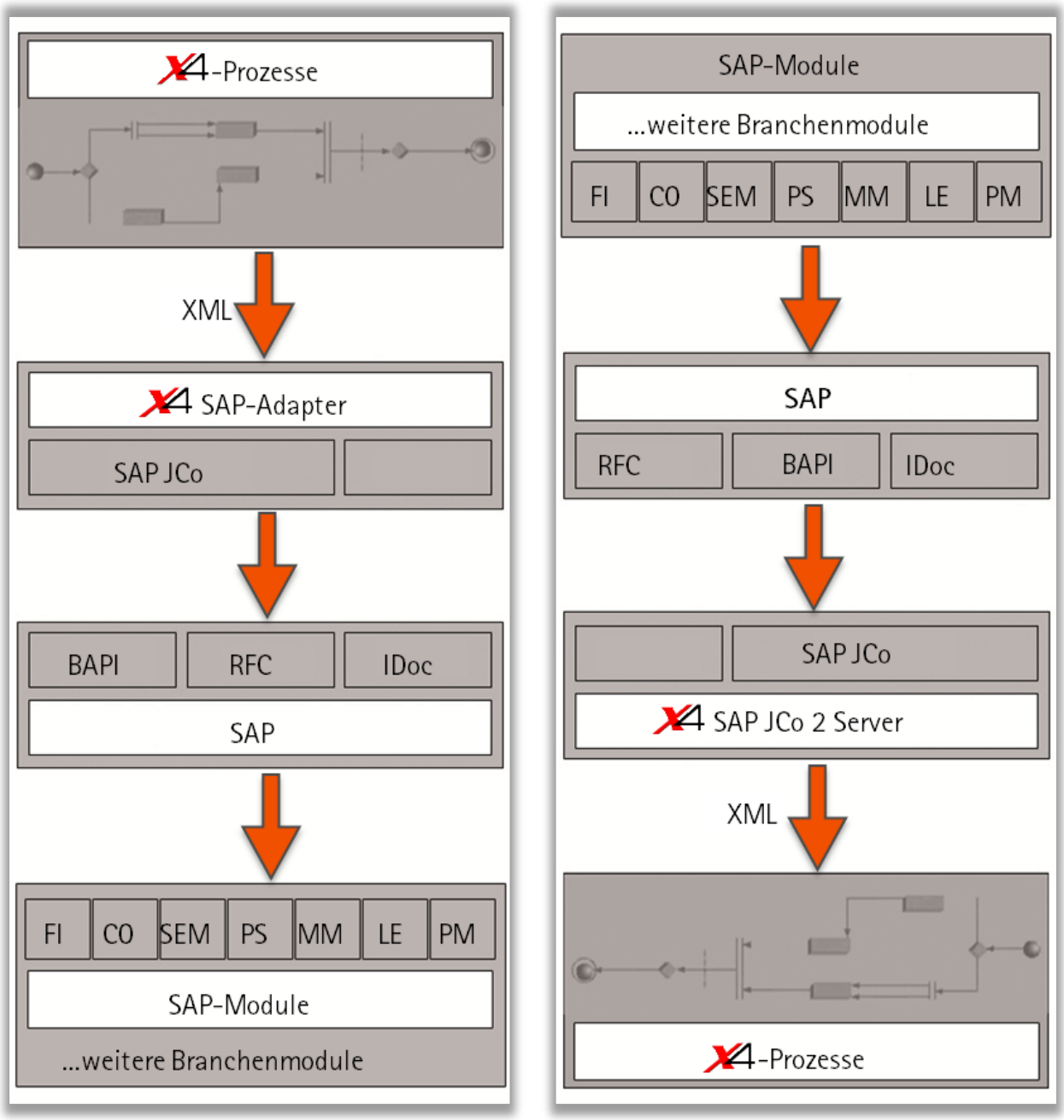
10.9.2.7 SAP Listener (JCo 3)

Der *SAP Listener (JCo 3)* ermöglicht die Verarbeitung von eingehenden Kommunikationsanfragen via BAPI/RFC und von IDocs eines SAP-Systems. Er stellt dabei automatisch ggf. unterbrochene Verbindungen wieder her.
Ehemalige Bezeichnung bis Version 4.5.1: *X4 SAP JCo 3 Listener*

Funktionen

Der *X4 Server* ermöglicht die bidirektionale Anbindung von *SAP R/3*-Systemen über den *SAP Java Connector* in Version 3.x (*JCo 3*). Dabei sind folgende Komponenten relevant:

- *SAP JCo 3 Server*: Ermöglicht die Verarbeitung von eingehenden Kommunikationsanfragen via BAPI/RFC und von IDocs eines SAP-Systems an den *X4 Server*. Der *SAP JCo 3 Server* stellt dabei automatisch ggf. unterbrochene Verbindungen wieder her.
- *SAP Connector (JCo 3)*: In Technical Processes kann dieser Funktions-Adapter einen beliebigen Remote Function Call (RFC) bzw. einen beliebigen BAPI-Funktionsbaustein in einem SAP-System aufrufen. Zudem können IDocs an *SAP R/3* versandt werden; siehe *Adapter und Bausteine, SAP Connector (JCo 3)*.
- *SAP Explorer (JCo 3)*: Dieser ermöglicht während der Prozess-Entwicklung das Erzeugen von Strukturbeschreibungen von IDocs und BAPIs. Zudem listet er BAPIs oder IDocs oder erzeugt leere IDoc- und BAPI-Strukturen und gibt je nach Operation ein entsprechendes XML-Dokument aus; siehe *Adapter und Bausteine, SAP Explorer (JCo 3)*.



Multi-SAP-Fähigkeit

Über den *JCo 3 Server* können mehrere SAP-Systeme über eine jeweils separate *JCo 3 Server*-Instanz angebunden werden. Zum Verwenden des *JCo 3*-Servers ist eine entsprechende Konfiguration erforderlich:

- *X4*-seitige Konfiguration in `X4config.xml`; siehe SAP JCo 3 Server-Konfiguration (X4config.xml)
- Definition von Verbindungs-Eigenschaften; siehe JCo3 Properties
- Zuordnung der Funktionsbaustein-Aufrufe zu Technical Processes und Commit/Rollback-Handler; siehe Funktionsbaustein-/Methodendefinitionen (X4 SAP JCo 3)

Identifikation von JCo 3-Servern über ProgID

Jede *JCo 3 Server*-Instanz bzw. jedes an SAP angebundene System kommuniziert über eine eindeutige *ProgID*, die in der Server-Properties-Datei, standardmäßig `jco_server_settings.prop`, konfiguriert wird und beim Start der Server-Instanz dem SAP-System bekannt gemacht wird.

Pro SAP-System können mehrere ProgIDs gleichzeitig verwendet werden. Über die ProgID werden IDocs und RFC/BAPIs werden an das jeweilige Ziel-System eindeutig "adressiert".

10.9.2.7.1 Installation von X4 SAP JCo 3

Welche Schritte zur Installation von *X4 SAP JCo 3* erforderlich sind

> **Beispiel**
> Als Beispiel wird eine Installation für den *JBoss* Applikations-Server auf Basis eines 32-Bit-Windows-Systems beschrieben

10.9.2.7.1.1 Voraussetzungen

Zur Installation eines *SAP JCo 3 Servers* im *X4 Server* werden *SAP*-Bibliotheken benötigt, die z.T. betriebssystem- und architekturabhängig sind. Diese Bibliotheken werden auch vom den *SAP Connector (JCo 3)* und dem *SAP Explorer (JCo 3)* benötigt.

Als registrierter SAP-Partner die erforderlichen Bibliotheken sowie weiterführende Dokumentation über das *SAPNet* von http://service.sap.com/connectors herunterladen.

> Wählen Sie hierbei zueinander kompatible Versionsnummern der Bibliotheken (diese sind zudem betriebssystem- und architekturabhängig).
> Achten Sie darauf, dass das verwendete Betriebsystem kompatibel mit der eingesetzten JDK-Version und der Version Ihres SAP-Systems ist. Folgende Matrix zeigt den Zusammenhang dieser drei Variablen.

		OS		**SAP**	
		32 bit	64 bit	32 bit	64 bit
JDK	32 bit	x	x	x	-
	64 bit	-	x	-	x

10.9.2.7.1.2 Bibliotheken installieren

1. Native Bibliotheken (je nach System) in den entsprechenden Ordner des Applikations-Servers für native Binär-Abhängigkeiten oder in das `bin`-Verzeichnis von Java kopieren.
   Beispiel: Unter Windows 32-Bit: `librfc32.dll`, `msvcr71.dll`, `msvcp71.dll` und `sapjcorfc.dll` nach `C:\Windows\System32` oder nach `C:\X4\Server\jdk1.7\bin` kopieren.
2. Folgende Java-Bibliotheken in den Classpath des Applikations-Servers kopieren:
   - `sapidoc.jar`
   - `sapjco.jar`

- `sapidocjco.jar`

Beispiel JBoss 4.2.3.GA: Diese Bibliotheken nach `C:\X4\JBoss4.2.3.GA\server\default\lib` kopieren.
Beispiel JBoss AS 7.1.1: Diese Bibliotheken unter `C:\X4\Server\jboss-as-7.1.1.Final\standalone\deployments` in das `lib`-Verzeichnis von `X4.ear` kopieren.

3. *X4 Server* neu starten.

10.9.2.7.1.3 Installation prüfen

Verbindung prüfen über den *SAP Explorer (JCo 3)* (analog zu *X4 Experience*-Beispiel `X4Experience/AdapterSuites/ X4SAPSuite/SAP(JCo2)Explorer(Prototype)/ Processes/CreateEmptyBapi.wrf`):

- Sicherstellen, dass die Konfiguration des *SAP JCo 3 Server* vollständig ist; siehe Konfiguration von X4 SAP JCo 3
- Adapterparameter `SystemType` auf *3* setzen (für eine Verbindung zu *SAP R/3*).
- Sicherstellen, dass Adapteroperation `Create Empty Bapi` gewählt ist.
- Prozess mit dem Input-XML `<Bapiname>BAPI's name</Bapiname>` ausführen und Ergebnis-XML prüfen.

Auf dieser Seite:

- Voraussetzungen
- Bibliotheken installieren
- Installation prüfen

Siehe auch:

- Konfiguration von X4 SAP JCo 3

10.9.2.7.2 Konfiguration von X4 SAP JCo 3

Welche Konfiguration *X4*-seitig angepasst werden muss, um den *JCo 3 Server* zu starten, von außen erreichbar zu machen und um den eingehenden RFC/BAPIs und IDocs entsprechende Handler-Prozesse im *X4 Server* zuzuordnen

10.9.2.7.2.1 SAP JCo 3 Server-Konfiguration (X4config.xml)

Wie der *X4 SAP JCo 3 Server* in `X4config.xml` konfiguriert wird

> ⓘ Die zentrale Konfigurationsdatei `<X4>\X4config.xml` enthält für den *X4 SAP JCo 3 Server* keine Konfigurationsoptionen mehr, sondern ggf. lediglich für eine separate *X4 SAP JCo 2*-Installation. Stattdessen befindet sich die *JCo 3*-Konfiguration in Properties-Dateien, die über die Adapter und/oder über die Listener-Konfiguration aufgerufen werden.

Siehe auch:

- JCo 3 Properties
- JCo 3 Listener-Konfiguration
- Konfiguration über X4config.xml

10.9.2.7.2.2 JCo 3 Listener-Konfiguration

Wie Sie eingehenden *JCo 3*-Verbindungen Properties und Methoden zuweisen

Die Konfiguration eingehender *JCo 3*-Verbindungen von einem oder mehreren SAP-Systemen befindet sich in Datei `saplistener.xml` in `<X4>\X4DB\0`. Dort lässt sich pro SAP-System ein *JCo 3*-Listener definieren, der wiederum Server-Properties und Methodendefinitionen (in einem spezifischen XML-Dokument) referenziert.

XML-Struktur
Zur Beschreibung der Struktur von `saplistener.xml` ist zudem ein XML-Schema-Definition `saplistener.xsd` im selben Ordner verfügbar. Diese XSD ist im Wurzel-Element `<listeners>` der Listener-Konfiguration angegeben.

```
<listener>
   <serverPropertiesURL>Pfad zur Properties-Datei</serverPropertiesURL>
   <methodDefinitionsURL>Pfad zur Methodendefinition</methodDefinitionsURL>
   <replaceEscapedControlChars>false</replaceEscapedControlChars><!-- true | false --
>
</listener>
```

Jedes Element `<listener>` mit einer *JCo 3*-Listener enthält folgende Kindelemente:

- `<serverPropertiesURL>`: Gültige URL zu einer Properties-Textdatei, die sämtliche Verbindungseinstellungen zum SAP-Host enthält
- `<methodDefinitionsURL>`: Gültige URL zu einem XML-Dokument, in dem sämtliche Methodendefinitionen bzw. Funktionsbausteinaufrufe und Transaktions-Handling-Prozesse definiert sind; siehe Funktionsbaustein-/Methodendefinitionen
- `<replaceEscapedControlChars>`: Maskierte Steuerzeichen (z.B.: *&#31;*) in den von der JCo-Bibliothek zurückgegebenen XML-Strings durch das Ersetzungszeichen ( ) austauschen. Mögliche Werte sind `true` oder `false`

Beispiel
Im folgenden Beispiel werden zur Anbindung von zwei SAP-Hosts jeweils ein Listener mit den separaten Properties- und Methodendefinitionen bestimmt. Mit `replaceEscapedControlChars` *true* werden zudem in den von der JCo-Bibliothek zurückgegebenen XML-Strings alle maskierten Steuerzeichen (z.B.: *&#31;*) durch das Ersetzungszeichen ( ) ausgetauscht.

```
<listeners xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:noNamespaceSchemaLocation="saplistener.xsd">
  <listener>
     <serverPropertiesURL>file:///C:/X4/X4DB/0/host1.prop</serverPropertiesURL>
     <methodDefinitionsURL>file:///C:/X4/X4DB/0/host1_method_def.xml
     </methodDefinitionsURL>
     <replaceEscapedControlChars>true</replaceEscapedControlChars>
  </listener>
  <listener>
     <serverPropertiesURL>file:///C:/X4/X4DB/0/host2.prop</serverPropertiesURL>
     <methodDefinitionsURL>file:///C:/X4/X4DB/0/host2_method_def.xml
     </methodDefinitionsURL>
     <replaceEscapedControlChars>true</replaceEscapedControlChars>
  </listener>
</listeners>
```

Auf dieser Seite:

- XML-Struktur
- Beispiel

Siehe auch:

- JCo 3 Properties
- Funktionsbaustein-/Methodendefinitionen (X4 SAP JCo 3)

10.9.2.7.2.3 JCo 3 Properties

Wie Sie die Verbindungsparameter zur Kommunikation zwischen Client und Host hinterlegen
Die Verbindungsdaten zur Kommunikation zwischen Client und Host können Sie global in einer Properties-Datei definieren. Für eingehende Verbindungen wird eine solche Properties-Datei pro *X4 SAP JCo 3*-Listener in `<X4>/X4DB/0/saplistener.xml` angegeben.

Für ausgehende Verbindungen über den *SAP Connector (JCo 3)* und den *SAP Explorer (JCo 3)* geben Sie den Pfad zu einer solchen Properties-Datei jeweils in Adapter-Parameter `PropertyFile` an.

Im Adapter gesetzte Parameterwerte überschreiben zur Laufzeit die Konfiguration in der Properties-Datei. Alle Properties, die in der Properties-Datei gesetzt werden (auch zusätzliche Properties), werden vom *X4 SAP JCo 3*-System übernommen.

Properties-Struktur

```
# connection String to SAP
jco.server.gwhost  = Gateway-Host
jco.server.gwserv  = Gateway-Service (normalerweise sapgwNN)
jco.server.progid  = Programm-ID
jco.server.trace   = RFC Trace aktivieren (1) oder deaktivieren (0)
jco.server.unicode = Im Unicode-Modus verbinden (1) oder nicht (0)

# equivalent to (dynamic) adapter parameter 'User'
jco.client.user = Benutzername

# equivalent to (dynamic) adapter parameter 'Password'
jco.client.passwd = Passwort

# equivalent to (dynamic) adapter parameter 'Client'
jco.client.client = SAP-Mandant (dreistelliger numerischer Wert)

# equivalent to (dynamic) adapter parameter 'Language'
jco.client.lang = Anmeldesprache

# equivalent to (dynamic) adapter parameter 'SystemNumber'
jco.client.sysnr = SAP-System-Nummer

# equivalent to (dynamic) adapter parameter 'RouterString'
jco.client.ashost = Router-String

# (optional) type of SAP system 2 = R/2, 3 = R/3, E = External

# equivalent to (dynamic) adapter parameter 'SystemType'
jco.client.type = Art des entfernten Hosts

# set to true if transactions are used
# equivalent to (dynamic) adapter parameter 'TransactionEnabled'
x4.transactionEnabled = Transaktionale Verarbeitung aktivieren (true)
oder deaktivieren (false)

# Should be set for each initiated transaction
# equivalent to (dynamic) adapter parameter 'TransactionId'
x4.transactionID = Transaktions-ID (für jede initiierte Transaktion)
```

ⓘ Eine vollständige Beschreibung aller (Verbindungs-) Parameter finden Sie in der *SAP JCo 3*-Dokumentation. Sämtliche Parameter werden vom *X4 SAP JCo 3 Server* unterstützt.

Beispiel
Im Folgenden ein Beispiel für eine Properties-Datei mit einigen gesetzten Werten:

```
# equivalent to (dynamic) adapter parameter 'User'
jco.client.user = <user>

# equivalent to (dynamic) adapter parameter 'Password'
jco.client.passwd = <pwd>

# equivalent to (dynamic) adapter parameter 'Client'
jco.client.client = 800

# equivalent to (dynamic) adapter parameter 'Language'
jco.client.lang = DE

# equivalent to (dynamic) adapter parameter 'SystemNumber'
jco.client.sysnr = 00

# equivalent to (dynamic) adapter parameter 'RouterString'
jco.client.ashost =

# (optional) type of SAP system  2 = R/2, 3 = R/3, E = External
# equivalent to (dynamic) adapter parameter 'SystemType'
jco.client.type = 3

# set to true if transactions are used
# equivalent to (dynamic) adapter parameter 'TransactionEnabled'
x4.transactionEnabled = false

# Should be set for each initiated transaction
# equivalent to (dynamic) adapter parameter 'TransactionId'
x4.transactionID = 1
```

Auf dieser Seite:

- Properties-Struktur
- Beispiel

Siehe auch:

- JCo 3 Listener-Konfiguration
- Funktionsbaustein-/Methodendefinitionen (X4 SAP JCo 3)

10.9.2.7.2.4 Funktionsbaustein-/Methodendefinitionen (X4 SAP JCo 3)

Wie Sie im SAP-Listener zu RFC/BAPIs jeweils einen verarbeitenden *X4*-Prozess zuordnen können In einem spezifischen XML-Dokument (z. B. `<X4>/X4DB/0/jco_method_def.xml`) können Sie für jeden Funktionsbaustein (Methode), der vom SAP-Host angesprochen wird, über den *JCo 3 Server* einen entsprechenden *X4*-Prozess zur Weiterverarbeitung zuordnen.

XML-Struktur für die SAP-Listener-Konfiguration

Die SAP-Listener-Konfigurationsdatei `<X4><X4>/X4DB/0/saplistener.xml` referenziert für jede *JCo 3 Server*-Instanz jeweils ein XML-Dokument mit Funktionsbaustein-/Methodendefinitionen. Darin können Sie innerhalb des Wurzel-Elements `<definitions>` beliebig viele Elemente `<method>` nach folgendem Muster anlegen:

```
<method>
  <name>Funktionsbaustein-/Methodenname</name>
    <wait_for_result>Boolescher Wert</wait_for_result>
    <class>de.softproject.integration.sapjcoserver.requesthandlers.
           WorkflowStarter</class>
    <properties>
      <property name="workflowname">Pfad zum X4-Prozess</property>
      <property name="userID">Benutzerordner in X4DB</property>
      <property name="raw">Boolescher Wert</property>
    </properties>
</method>
```

Für jede Funktionsbaustein-/Methodendefinition gilt:

- Element `<name>` enthält den Namen des durch SAP aufgerufenen Funktionsbaustein (Methode); Der Platzhalter `*` steht für alle anderen Funktionsbausteine , für die kein expliziter Funktionsbaustein definiert wurde. Zudem lassen sich hier Transaktions-Handler definieren (siehe unten).
- In Element `<wait_for_result>` können Sie konfigurieren, ob SAP auf die Antwort warten soll (*`false`*), oder ob der Funktionsbaustein-/Methodenaufruf asynchron erfolgt (*`true`*).
- Element `<class>` gibt die verwendete Klasse an, die zur Verarbeitung des Funktionsbausteins (bzw. der Methode) hinterlegt ist (normalerweise ist dies `de.softproject.integration.sapjcoserver.requesthandlers.WorkflowStarter` zum Starten von *X4*-Prozessen); in den Kindelementen von `<class>` geben Sie die zugehörigen Eigenschaften in jeweils einem Element `<property>` an.

Für die *X4*-Prozess-Starterklasse werden folgende Eigenschaften angegeben:

- `workflowname`: Pfad zum *X4*-Prozess, der die Verarbeitung des Funktionsbaustein-/Methodenaufrufs übernimmt
- `userID`: Repository-Benutzerordner, in dem der *X4*-Prozess liegt
- `raw`: IDoc im Raw-Format verarbeiten (*`true`*) oder (*`false`*)

Kompatibilitätshinweis
In älteren Funktionsbaustein-/Methodendefinitionen finden sich ggf. zusätzliche Angaben in einem Element `<type>` und in einem Element `<property name="languageID">`. Diese besitzen keine Funktion und können daher weggelassen werden.

> Beachten Sie bei der Migration auf JCo 3, dass unter JCo 2 das Attribut `name` innerhalb des Elementes `<property>` groß geschrieben wurde. Unter JCo 3 wird dieses klein geschrieben.

Transaktions-Handler für IDocs und BAPI-Aufrufe

In *X4 SAP JCo 3* wurde das Transaktions-Handling in die Methodendefinition integriert. Zur Verarbeitung von eingehenden IDocs und BAPI-Aufrufen können Sie wie folgt einen *Commit*-Handler und einen *Rollback*-Handler definieren und jeweils einen *X4*-Prozess zuordnen, indem Sie in jeweils einer der Methodendefinition innerhalb von Element `<name>` folgende Werte hinterlegen:

- *`IDOC+COMMIT`*: IDoc-Transaktion abschließen (*Commit*)
- *`IDOC+ROLLBACK`*: IDoc-Transaktion abbrechen und zurücksetzen (*Rollback*)
- *`BAPI+COMMIT`*: BAPI-Transaktion abschließen (*Commit*)
- *`BAPI+ROLLBACK`*: BAPI-Transaktion abbrechen und zurücksetzen (*Rollback*)

Globaler Handler
Falls kein spezifischer Handler zutrifft, können Sie einen globalen Handler definieren, indem Sie in der Methodendefinition in Element `<name>` als Wert einen Stern (`*`) eintragen.

Beispiel
Im folgenden Beispiel wird für den Funktionsbaustein (Methode) `ORDERS05` der *X4*-Prozess `save_idoc.wrf` aufgerufen und es werden Daten im *IDoc Raw*-Format für die nachgeordnete Verarbeitung bereitgestellt. Zudem wird ein BAPI-Commit-Handler sowie ein globaler Handler ohne zugeordneten *X4*-Prozess definiert.

```
<method>
  <name>ORDERS05</name>
    <wait_for_result>false</wait_for_result>
    <class>de.softproject.integration.sapjcoserver.requesthandlers.
           WorkflowStarter</class>
    <properties>
      <property Name="workflowname">Project/Folder/ProcessIDoc.wrf</property>
      <property Name="userID">1</property>
      <property Name="raw">true</property>
    </properties>
</method>
<method>
      <name>BAPI+COMMIT</name>
      <type>0</type>
      <wait_for_result>false</wait_for_result>
      <class>de.softproject.integration.sapjco3server.requesthandlers.
             WorkflowStarter</class>
      <properties>
         <property name="workflowname">Project/Folder/BAPI_Commit.wrf</property>
         <property name="userID">1</property>
      </properties>
</method>
<method>
      <name>*</name>
      <type>1</type>
      <wait_for_result>false</wait_for_result>
      <class>de.softproject.integration.sapjco3server.requesthandlers.
             WorkflowStarter</class>
      <properties>
      </properties>
</method>
```

Auf dieser Seite:

- XML-Struktur für die SAP-Listener-Konfiguration
- Kompatibilitätshinweis
- Transaktions-Handler für IDocs und BAPI-Aufrufe
- Globaler Handler
- Beispiel

10.10 File Processing

- ASCII to XML
- Base64 Batch Decoder
- Base64 Converter
- Charset Changer
- CSV
- Encoding Converter

- Hex Converter
- Image Processing
- Java Properties File Reader
- QR Code Generator
- Regular Expression
- Stream Splitter
- Structured Text
- Text Replacement
- Transliteration Adapter

10.10.1 ASCII to XML

10.10.1.1 ASCII to XML Converter

`ASCII to XML Converter`: Liest ein Textdokument ein und gibt ein XML-Dokument mit unveränderter Zeichenkodierung aus, das den Inhalt des Textdokuments mit einem CDATA-Bereich umschließt.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`WrapData`*: Text-Dokument in ein XML-Dokument mit CDATA-Bereich umwandeln

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.ASCII2XML`*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.10.1.2 ASCII Writer

`ASCII Writer`: Konvertiert ein XML-Dokument gemäß den Regeln einer Deskriptordatei in ein strukturiertes Textdokument mit unveränderter Zeichenkodierung.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`Execute`*: ASCII-Dokument schreiben, gemäß den Regeln einer in Parameter `DescFile` festgelegten ASCII-Deskriptor-Datei

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.asciiwriter.ASCIIWriter*: Hauptklasse (Standard)
DescFile	Name der Deskriptordatei, die das Textdokument in seiner Datenstruktur beschreibt. Die Deskriptordatei sollte im Benutzerordner des X4 Repositorys abgelegt sein (z. B. `X4DB/1/Project/Descr.ad`) **Mögliche Werte:** Pfad und Dateiname (z. B. */Project/Descr.ad*)

Statuswerte

-1	Beim Ausführen des Adapters trat ein Fehler auf (siehe *X4 Server*-Protokoll).
0	Der Adapter wurde erfolgreich ausgeführt.

10.10.1.3 Text to Lines Converter

`Text to Lines Converter`: Liest alle Zeilen eines Textdokuments oder bestimmte Zeilen ein und gibt ein XML-Dokument aus, das für jede Zeile ein XML-Element enthält.
Als Input verwendet der Adapter ein Textdokument aus dem vorhergehenden Prozess-Schritt. Mit Operation `Iterate` kann der Adapter das Input-Dokument schrittweise durchlaufen und für jede Iteration im Technical Process wird ein Ergebnis-Dokument ausgegeben.

Beachten Sie:

- Wenn Sie innerhalb eines Prozess-Kontextes mehrere *Text to Lines Converter* definieren, dann vergeben Sie in Parameter id eine eindeutige ID für den jeweiligen Adapter, z.B. *1001*, *1002* etc. So wird für das Selektieren im Iterator-Modus und für das Zurücksetzen eines Iterators gewährleistet, dass nur der *Text to Lines Converter Adapter* mit der entsprechenden ID angesprochen wird (siehe Operation *Clear*).
- Wenn Sie in Parameter `id` den Standardwert *0* belassen, dann wird die Nummer des Prozess-Schritts (*Action ID*) im aktuellen Prozess als ID verwendet. Diese ID ist möglicherweise nicht eindeutig, wenn sich in einem Subprozess ein *Text to Lines Converter* mit der gleichen Action-ID befindet.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *Parse*: Zeilen des Textdokuments einlesen • *Iterate*: Iterator-Modus aktivieren und Textdokument zeilenweise einlesen • *Clear*: Iterator eines *Text to Lines Converter Adapters* zurücksetzen, der dieselbe ID besitzt wie der gewünschte *Text to Lines Converter Adapter* mit der Operation *Iterate* (siehe Parameter `id`)

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.text2lines.TextToLinesAdapter*: Hauptklasse (Standard)
start	Erste Zeile, die eingelesen werden soll (Zählweise beginnt bei 0) **Mögliche Werte:** • Beliebige ganze Zahl • *0*: Ab einschließlich der ersten Zeile einlesen (Standard)

count	Anzahl der Zeilen, die eingelesen werden sollen (bei Operation Iterate pro Durchgang) **Mögliche Werte:** • Beliebige positive ganze Zahl • *0*: Keine Beschränkung, d. h. alle Zeilen auf einmal einlesen (Standard)
id	Eindeutige ID des *Text to Lines Converter Adapter*-Bausteins innerhalb des Prozess-Kontextes bzw. der Prozess-Instanz (wenn Sie mehrere voneinander unabhängig iterierende *Text to Lines Converter Adapter* in demselben Prozess-Kontext verwenden möchten) **Mögliche Werte:** • *0*: Aktuelle Action-ID (Nummer des Prozess-Schritts) als ID verwenden; Beachten Sie hierbei, dass ein *Text to Lines Converter Adapter* in einem Subprozess die gleiche Action-ID besitzen könnte. (Standard) • Beliebige ganze Zahl außer *0*, um den jeweiligen *Text to Lines Converter Adapter*-Baustein innerhalb des Prozess-Kontextes eindeutig zu kennzeichnen; Empfehlung: einen Wert *>1000* verwenden, um nicht zufällig eine bereits bestehende Action-ID zu "erwischen"

Ausgabe

Dieser Adapter gibt ein UTF-8-kodiertes XML-Dokument mit folgender Struktur aus. Für jede Zeile wird ein XML-Element `<Line/>` erzeugt, das die Zeilennummer und den Inhalt der jeweiligen Zeile enthält:

```
<Text>
  <Line lineNo="Zeilennummer"><!--Inhalt der Zeile--></Line>
</Text>
```

Statuswerte

-1	Fehler bei der Ausführung des Adapters (Details siehe Server-Log)
0	Der Adapter gibt keine Daten (mehr) aus. Dieser Status kann bei Operation `Iterate` als Abbruchbedingung einer Schleife im Technical Process ausgewertet werden und gibt an, dass keine weiteren Datensätze verfügbar sind.

1	Der Adapter gibt ein Ergebnis aus. Dieser Status kann bei Operation `Iterate` als Schleifeneintrittsbedingung im Technical Process ausgewertet werden und gibt an, dass ein weiterer Datensatz verfügbar ist.

10.10.1.4 Text to Lines Converter from URL

`Text to Lines Converter from URL`: Liest Zeilen eines Textdokuments von einer beliebigen gültigen URL ein und gibt ein XML-Dokument aus, das die gewünschten Zeilen enthält. Mit der Operation `Iterate` kann der Adapter das Input-Dokument schrittweise durchlaufen und für jede Iteration im Technical Process wird ein Ergebnis-Dokument ausgegeben.

> Wenn Sie innerhalb eines Prozess-Kontextes mehrere *Text to Lines Converter from URL* definieren, dann vergeben Sie in Parameter id eine eindeutige ID für den jeweiligen Adapter, z.B. *`1001`*, *`1002`* etc. So wird für das Selektieren im Iterator-Modus und für das Zurücksetzen eines Iterators gewährleistet, dass nur der entsprechende Adapter angesprochen wird (siehe Operation *`Clear`*).
> Wenn Sie in Parameter `id` den Standardwert *`0`* belassen, dann wird die Nummer des Prozess-Schritts (*Action ID*) im aktuellen Prozess als ID verwendet. Diese ID ist möglicherweise nicht eindeutig, wenn sich in einem Subprozess ein *Text to Lines Converter from URL* mit der gleichen Action-ID befindet.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`Parse`*: Gesamtes Textdokument einlesen • *`Iterate`*: Iterator-Modus aktivieren und Textdokument zeilenweise einlesen • *`Clear`*: Iterator eines *Text to Lines Converter from URL Adapters* zurücksetzen, der dieselbe ID besitzt wie der gewünschte *Text to Lines Converter from URL* mit der Operation *`Parse`* (siehe Parameter id)

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.text2lines.TextToLinesUrlAdapter`*: Hauptklasse (Standard)

start	Erste Zeile, die eingelesen werden soll (Zählweise beginnt bei 0) **Mögliche Werte:** • Beliebige ganze Zahl • *0*: Ab einschließlich der ersten Zeile einlesen (Standard)
count	Anzahl der Zeilen, die eingelesen werden sollen (bei Operation Iterate pro Durchgang) **Mögliche Werte:** • Beliebige positive ganze Zahl • *0*: Keine Beschränkung, d. h. alle Zeilen auf einmal einlesen (Standard)
id	Eindeutige ID des *Text to Lines Converter from URL*-Bausteins innerhalb des Prozess-Kontextes bzw. der Prozess-Instanz (wenn Sie mehrere voneinander unabhängig iterierende *Text to Lines Converter from URL* in demselben Prozess-Kontext verwenden möchten) **Mögliche Werte:** • *0*: Aktuelle Action-ID (Nummer des Prozess-Schritts) als ID verwenden; Beachten Sie hierbei, dass ein *Text to Lines Converter from URL* in einem Subprozess die gleiche Action-ID besitzen könnte. (Standard) • Beliebige ganze Zahl außer *0*, um den jeweiligen *Text to Lines Converter from URL*-Baustein innerhalb des Prozess-Kontextes eindeutig zu kennzeichnen; Empfehlung: einen Wert *>1000* verwenden, um nicht zufällig eine bereits bestehende Action-ID zu "erwischen"
url	URL zum einzulesenden Textdokument **Mögliche Werte:** Beliebige gültige URL (z.B. *xstore://Project/Folder/File.txt*)
encoding	Zeichenkodierung des Input-Textdokuments **Mögliche Werte:** • Beliebige gültige Zeichenkodierung (z.B. *UTF-8*) • *(leer)*: Standard-Zeichenkodierung des Systems verwenden

Ausgabe

Dieser Adapter gibt ein UTF-8-kodiertes XML-Dokument mit folgender Struktur aus. Für jede Zeile wird ein XML-Element `<Line/>` erzeugt, das die Zeilennummer und den Inhalt der jeweiligen Zeile enthält:

```
<Text>
  <Line lineNo="Zeilennummer"><!--Inhalt der Zeile--></Line>
</Text>
```

Beispiel

Das Beispiel `TextToLinesConverterFromURL.wrf` für den Iterator-Modus sowie für den einfachen Einlesemodus finden Sie im Demosystem *X4 Experience*.

Statuswerte

-1	Fehler bei der Ausführung des Adapters (Details siehe Server-Log)
0	Der Adapter gibt keine Daten (mehr) aus. Dieser Status kann im Iterator-Modus als Abbruchbedingung einer Schleife im Technical Process ausgewertet werden und gibt an, dass keine weiteren Datensätze verfügbar sind.
1	Der Adapter gibt ein Ergebnisdokument aus. Dieser Status kann im Iterator-Modus als Schleifeneintrittsbedingung im Technical Process ausgewertet werden und gibt an, dass ein weiterer Datensatz verfügbar ist.

10.10.2 Base64 Batch Decoder

`Base64 Batch Decoder`: Dekodiert Base64-kodierte Daten aus einer adapterspezifischen XML-Struktur, teilt das Ergebnis in separate Dateien auf und speichert diese in einem Ordner.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *Decode*: Daten aus dem XML-Dokument in Base64-kodierte Dateien schreiben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.docadapter.DocAdapter*: Hauptklasse (Standard)

directory	Zielordner, in den die dekodierten Dateien gespeichert werden sollen **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *%USER_COLLECTION%\Project\batch*)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Adapter erwartet eine adapterspezifische Input-XML-Struktur. Diese enthält für jede Datei, die dekodiert werden soll, den Dateinamen und den Base64-kodierten Dateiinhalt. Jedes Element `<document/>` steht hierbei für eine zu dekodierende Datei.

```
<data>
    <documents>
        <document>
            <filename><!--Dateiname--></filename>
            <content><!--Base64-kodierte Daten--></content>
        </document>
    </documents>
</data>
```

10.10.3 Base64 Converter

`Base64 Converter`: Enkodiert oder dekodiert Daten in Base64-Kodierung. Beim Dekodieren unterscheidet der Adapter anhand des MIME-Typs: ist der MIME-Typ `text/xml` oder `application/xml`, dann wird das Input-Dokument als XML-Dokument interpretiert und der Textinhalt des Wurzel-Elements dekodiert, ansonsten wird das komplette Dokument als Base64-kodierte Zeichenkette interpretiert.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *Encode*: Daten Base64-enkodieren • *Decode*: Daten in Base64-Kodierung dekodieren (abhängig vom MIME-Typ)

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.Base64Adapter`*: Hauptklasse (Standard)
`chunk`	Ausgabe in Blöcke mit einer Länge von 76 Zeichen unterteilen **Mögliche Werte:** • *`true`*: Zeichenkette in Zeilenblöcke unterteilen • *`false`*: Zeichenkette nicht unterteilen (Standard)
`encoding`	Verwendete Zeichenkodierung **Mögliche Werte:** Zeichenkette mit der verwendeten Zeichenkodierung, z. B. `UTF-8` (Standard), `windows-1251`, `ISO 8859-1`, ...

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.10.4 Charset Changer

`Charset Changer`: Ändert die Zeichenkodierung eines Dokuments.

> Die Zeichen des Input-Dokuments werden nicht in den Ziel-Zeichenkodierung konvertiert! Hierfür den Adapter *Encoding Converter* verwenden, siehe Encoding Converter.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`Execute`*: Zeichensatz der Datei ändern

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.CharsetChanger*: Hauptklasse (Standard)

InCharset	Zeichensatz des Input-Dokuments **Mögliche Werte:** • Beliebige in Java bekannte Zeichenkodierung (z. B. *Windows-1252*) • `UTF-8`: Unicode-Zeichenkodierung UTF-8 (Standard)
OutCharset	Zeichensatz des Ausgabe-Dokuments **Mögliche Werte:** • Beliebige in Java bekannte Zeichenkodierung (z. B. `ISO-8859-1`) • `UTF-8`: Unicode-Zeichenkodierung UTF-8

Statuswerte

`-1`	Die Operation wurde erfolgreich ausgeführt.
`1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.10.5 CSV

10.10.5.1 CSV Converter

`CSV Converter`: Liest ein CSV-Textdokument (*comma separated values*) aus dem vorhergehenden Prozess-Schritt ein und gibt für die angegebenen Datensätze ein XML-Dokument in UTF-8-Zeichenkodierung aus, das ggf. die Spaltenbezeichnungen als Elementnamen verwendet.

Zeilen vor Beginn der CSV-Datenzeilen sowie Kommentarzeilen können ausgelassen werden. Sowohl CR (*carriage return*) als auch CRLF (*carriage return line feed*) werden als Zeilenumbrüche interpretiert. Mit Operation `Iterate` kann der Adapter das Input-Dokument schrittweise durchlaufen und für jede Iteration im *X4*-Prozess wird ein Ergebnis-Dokument ausgegeben.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *Parse*: CSV-Textdokument in einem Schritt einlesen • *Iterate*: CSV-Textdokument schrittweise einlesen • *Clear*: Iterator eines *CSV Converters* zurücksetzen, der dieselbe ID besitzt wie der gewünschte *CSV Converter* mit der Operation *Iterate* (siehe Parameter `id`)

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.csv.CsvAdapter*: Hauptklasse (Standard)
`start`	Erste Datenzeile, die eingelesen werden soll (die Zählung beginnt bei 0; wenn die CSV-Datei eine Zeile mit Spaltenbezeichnungen enthält, dann wird diese ausgelassen.) **Mögliche Werte:** • Beliebige positive ganze Zahl • *0*: Alle Datenzeilen ab der ersten Zeile einlesen (Standard)
`count`	Anzahl der Datensätze, die eingelesen werden sollen (bei Operation Iterate pro Durchgang) **Mögliche Werte:** • Beliebige positive ganze Zahl • *0*: Keine Beschränkung, d. h. alle Datensätze auf einmal einlesen (Standard)

`id`	Eindeutige ID des *CSV Converter Adapter*-Bausteins innerhalb des Prozess-Kontextes oder der Prozess-Instanz (wenn Sie mehrere voneinander unabhängig iterierende *CSV Converter Adapter* in demselben Prozess-Kontext verwenden möchten) **Mögliche Werte:** • *0*: Aktuelle Action-ID (Nummer des Prozess-Schritts) als ID verwenden; Beachten Sie hierbei, dass ein *CSV Converter Adapter* in einem Subprozess die gleiche Action-ID besitzen könnte. (Standard) • Beliebige ganze Zahl außer *0*, um den jeweiligen *CSV Converter Adapter*-Baustein innerhalb des Prozess-Kontextes eindeutig zu kennzeichnen; Empfehlung: einen Wert *>1000* verwenden, um nicht zufällig eine bereits bestehende Action-ID zu verwenden.
`columnNamesInFirstRow`	Datenfelder der ersten Zeile in der CSV-Textdatei als Spaltenbezeichnungen interpretieren und Datenfeld-Elemente im XML-Dokument entsprechend benennen, falls es sich um gültige XML-Elementnamen handelt **Mögliche Werte:** • *Aktiviert*: Die erste Zeile im CSV-Textdokument enthält Spaltenbezeichnungen, die sich als gültige XML-Elementnamen (ohne Namespace-Angaben) eignen. • *Deaktiviert*: CSV-Textdokument beginnt direkt mit Datenzeilen; die Elemente werden <Column> oder <Column_n> (bei aktiviertem Parameter `useIndexedColumns`) benannt. (Standard)

`normalizeHeaderNames`	Gibt an, wie die Kopfzeilen normalisiert werden sollen, falls darin Zeichen enthalten sind, die nicht XML-konform sind, z. B. Leer- oder Sonderzeichen. **Mögliche Werte:** • *Special element name* (Standard): Name der Kopfzeile nicht anpassen, sondern durch `<Column>` ersetzen (Bsp: `author book` → `<Column>`) • *Skip non compliant characters*: Nicht-XML-konforme Zeichen überspringen (Bsp: `author book` → `<authorbook>`) • *Replace with dash*: Nicht-XML-konforme Zeichen werden durch – ersetzt (Bsp: `author book` → `<author-book>`) • *Replace with underscore*: Nicht-XML-konforme Zeichen werden durch _ ersetzt (Bsp: `author book` → `<author_book>`)
`columnNamesAfterStart`	Werte aus der Zeile, die in Parameter `start` angegeben wurde, als Spaltenbezeichnungen interpretieren und im XML-Dokument die Elemente entsprechend benennen, falls es sich um gültige XML-Elementnamen handelt **Mögliche Werte:** • *Aktiviert*: Die übersprungenen Reihen ignorieren und den Spaltenbezeichnungen aus der ersten verarbeiteten Zeile verwenden • *Deaktiviert*: Die erste verarbeitete Zeile enthält keine Spaltenbezeichnungen; die Elemente werden <Column> oder <Column_n> (bei aktiviertem Parameter `useIndexedColumns`) benannt. (Standard)
`delimiter`	Das Trennzeichen zwischen den einzelnen Datenfeldern einer Zeile im CSV-Textdokument **Mögliche Werte:** • Ein beliebiges Zeichen (keine Escape-Sequenzen wie *\t*) • *;*: Semikolon (Standard)
`rowName`	Elementname im ausgegebenen XML-Dokument, das jede Datenzeile umschließt **Mögliche Werte:** XML-Elementname (ohne Namespace)

`excel`	Bestimmt, ob das CSV-Textdokument von *MS Excel* erzeugt wurde (kein Standard-CSV-Format) und dementsprechend konvertiert werden muss **Mögliche Werte:** • *Aktiviert*: CSV-Textdokument im *MS Excel CSV*-Format verarbeiten • *Deaktiviert*: Standard-CSV-Textdokument verarbeiten (Standard)
`useIndexedColumns`	Den Datenelementnamen die entsprechende Spaltennummer (beginnend bei 0) hinzufügen, wenn keine Spaltenbezeichnungen verfügbar sind **Mögliche Werte:** • *Aktiviert*: Spaltennummer an den Elementnamen anhängen (z. B. *<Column_0>*) • *Deaktiviert*: Alle Datenelemente erhalten den Namen `<Column>` (Standard)
`commentStart`	Präfix zur Kennzeichnung von Kommentarzeilen; zutreffende Zeilen werden ausgelassen. **Mögliche Werte:** • Beliebige Zeichenkette (jedoch keine Trenn- oder Anführungszeichen) (z. B. `****`) • *(leer)*: Keine Kommentarbehandlung; Zeilen werden als Datensätze interpretiert. (Standard)
`inputEncoding`	Definiert, welche Zeichenkodierung im Adapter-Input verwendet wird. **Mögliche Werte:** • `UTF-8` • `windows-1252` • `ISO 8859-1` ⓘ Als Zeichenkodierung für den Adapter-Output wird immer UTF-8 verwendet.

Statuswerte

-1	Fehler bei der Ausführung des Adapters (Details siehe Server-Log)
0	Der Adapter gibt keine Daten (mehr) aus. Dieser Status kann bei Operation `Iterate` als Abbruchbedingung einer Schleife im *X4*-Prozess ausgewertet werden und gibt an, dass keine weiteren Datensätze verfügbar sind.
1	Der Adapter gibt ein Ergebnis aus. Dieser Status kann bei Operation `Iterate` als Schleifeneintrittsbedingung im *X4*-Prozess ausgewertet werden und gibt an, dass ein weiterer Datensatz verfügbar ist.

Input

Der Adapter erwartet eine beliebige strukturierte CSV-Textdatei (ggf. auch von *MS Excel* erzeugt) mit einem beliebigen Trennzeichen und optionalen Kommentarzeilen.

Ausgabe

Dieser Funktions-Adapter gibt ein UTF-8-kodiertes XML-Dokument mit folgender Struktur aus. Für jede verarbeitete Zeile wird ein Element erzeugt, das ggf. den Namen aus Parameter `rowName` enthält:

```
<Csv>
   <!-- Wenn Spaltenbezeichnungen übernommen werden -->
   <Elementname rowNo="Zeilennummer im Ausgangsdokument">
      <Spaltenbezeichnung>Wert</Spaltenbezeichnung>
      <!--...-->
   </Elementname>

   <!-- Wenn keine Spaltenbezeichnungen verfügbar sind -->
   <Elementname rowNo="Zeilennummer im Ausgangsdokument">
      <Column>Wert</Column>
      <!--...-->
   </Elementname>

   <!-- Wenn keine Spaltenbezeichnungen verfügbar sind
        und Parameter 'useIndexedColumns' aktiviert ist -->
   <Elementname rowNo="Zeilennummer im Ausgangsdokument">
      <Column_Spaltennummer>Wert</Column_Spaltennummer>
      <!--...-->
   </Elementname>
</Csv>
```

Beispiel

Eine CSV-Textdatei mit Kommentaren soll verarbeitet werden, die erst in der zweiten Zeile die Spaltenbeschreibungen enthält, beispielsweise:

```
**** This line is just a comment and should not be processed. ****
"METADATA","Value 1","Value 2","Value 3","Value 4","Value 5"
"CustomerID","CompanyName","ContactName","ContactTitle","Address","PostalCode"
"ALFKI","Alfreds Futterkiste","Maria Anders","Sales","Obere Str. 57","12209"
**** This line is just a comment and should not be processed. ****
"ANTON","Moreno Taquería","Antonio Moreno","Owner","Mataderos  2312","05023"
```

Hierzu wird der Adapter-Parameter `columnNamesAfterStart` aktiviert und Parameter `start` auf *1* gesetzt, weil sowohl Kommentar- als auch Spaltenbeschriftungszeile nicht mitgezählt werden. Mit dem Wert *Dataset*

für Parameter `rowName` wird beispielsweise folgendes XML-Dokument erzeugt:

```
<Csv>
  <Dataset rowNo="2">
    <CustomerID>ALFKI</CustomerID>
    <CompanyName>Alfreds Futterkiste</CompanyName>
    <ContactName>Maria Anders</ContactName>
    <ContactTitle>Sales</ContactTitle>
    <Address>Obere Str. 57</Address>
    <PostalCode>12209</PostalCode>
  </Dataset>
  <Dataset rowNo="3">
    <CustomerID>ANTON</CustomerID>
    <CompanyName>Moreno Taquería</CompanyName>
    <ContactName>Antonio Moreno</ContactName>
    <ContactTitle>Owner</ContactTitle>
    <Address>Mataderos 2312</Address>
    <PostalCode>05023</PostalCode>
  </Dataset>
</Csv>
```

10.10.5.2 CSV Converter from URL

`CSV Converter from URL`: Liest ein CSV-Textdokument (*comma separated values*) von einer beliebigen gültigen URL ein und gibt für die angegebenen Datensätze ein XML-Dokument in UTF-8-Zeichenkodierung aus, das ggf. die Spaltenbezeichnungen als Elementnamen verwendet.
Zeilen vor Beginn der CSV-Datenzeilen sowie Kommentarzeilen können ausgelassen werden. Sowohl CR (*carriage return*) als auch CRLF (*carriage return line feed*) werden als Zeilenumbrüche interpretiert. Mit Operation `Iterate` kann der Adapter das Input-Dokument schrittweise durchlaufen und für jede Iteration im Technical Process wird ein Ergebnis-Dokument ausgegeben.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`Parse`*: CSV-Textdokument in einem Schritt einlesen • *`Iterate`*: CSV-Textdokument schrittweise einlesen • *`Clear`*: Iterator eines *CSV Converter from URL Adapters* zurücksetzen, der dieselbe ID besitzt wie der gewünschte *CSV Converter from URL Adapters* mit der Operation *`Iterate`* (siehe Parameter `id`)

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.csv.CsvAdapter*: Hauptklasse (Standard)
`start`	Erste Datenzeile, die eingelesen werden soll (die Zählung beginnt bei 0; wenn die CSV-Datei eine Zeile mit Spaltenbezeichnungen enthält, dann wird diese ausgelassen.) **Mögliche Werte:** • Beliebige positive ganze Zahl • *0*: Alle Datenzeilen ab der ersten Zeile einlesen (Standard)
`count`	Anzahl der Datensätze, die eingelesen werden sollen (bei Operation `Iterate` pro Durchgang) **Mögliche Werte:** • Beliebige positive ganze Zahl • *0*: Keine Beschränkung, d. h. alle Datensätze auf einmal einlesen (Standard)

`id`	Eindeutige ID des *CSV Converter from URL Adapter*-Bausteins innerhalb des Prozess-Kontextes bzw. der Prozess-Instanz (wenn Sie mehrere voneinander unabhängig iterierende *CSV Converter from URL Adapter* in demselben Prozess-Kontext verwenden möchten) **Mögliche Werte:** • *0*: Aktuelle Action-ID (Nummer des Prozess-Schritts) als ID verwenden; Beachten Sie hierbei, dass ein *CSV Converter from URL Adapter* in einem Subprozess die gleiche Action-ID besitzen könnte. (Standard) • Beliebige ganze Zahl außer *0*, um den jeweiligen *CSV Converter from URL Adapter*-Baustein innerhalb des Prozess-Kontextes eindeutig zu kennzeichnen; Empfehlung: einen Wert *> 1000* verwenden, um nicht zufällig eine bereits bestehende Action-ID zu "erwischen"
`columnNamesInFirstRow`	Datenfelder der ersten Zeile in der CSV-Textdatei als Spaltenbezeichnungen interpretieren und Datenfeld-Elemente im XML-Dokument entsprechend benennen, falls es sich um gültige XML-Elementnamen handelt **Mögliche Werte:** • *Aktiviert*: Die erste Zeile im CSV-Textdokument enthält Spaltenbezeichnungen, die sich als gültige XML-Elementnamen (ohne Namespace-Angaben) eignen. • *Deaktiviert*: CSV-Textdokument beginnt direkt mit Datenzeilen; die Elemente werden `<Column>` oder `<Column_n>` (bei aktiviertem Parameter `useIndexedColumns`) benannt. (Standard)
`normalizeHeaderNames`	Gibt an, wie die Kopfzeilen normalisiert werden sollen, falls darin Zeichen enthalten sind, die nicht XML-konform sind, z. B. Leer- oder Sonderzeichen. **Mögliche Werte:** • *Special element name* (Standard): Name der Kopfzeile nicht anpassen, sondern durch `<Column>` ersetzen (Bsp: `author book` → `<Column>`) • *Skip non compliant characters*: Nicht-XML-konforme Zeichen überspringen (Bsp: `author book` → `<authorbook>`) • *Replace with dash*: Nicht-XML-konforme Zeichen werden durch – ersetzt (Bsp: `author book` → `<author-book>`) • *Replace with underscore*: Nicht-XML-konforme Zeichen werden durch _ ersetzt (Bsp: `author book` → `<author_book>`)

`columnNamesAfterStart`	Werte aus der Zeile, die in Parameter `start` angegeben wurde, als Spaltenbezeichnungen interpretieren und im XML-Dokument die Elemente entsprechend benennen, falls es sich um gültige XML-Elementnamen handelt **Mögliche Werte:** • *Aktiviert*: Die übersprungenen Reihen ignorieren und den Spaltenbezeichnungen aus der ersten verarbeiteten Zeile verwenden • *Deaktiviert*: Die erste verarbeitete Zeile enthält keine Spaltenbezeichnungen; die Elemente werden `<Column>` oder `<Column_n>` (bei aktiviertem Parameter `useIndexedColumns`) benannt. (Standard)
`delimiter`	Das Trennzeichen zwischen den einzelnen Datenfeldern einer Zeile im CSV-Textdokument **Mögliche Werte:** • Ein beliebiges Zeichen (keine Escape-Sequenzen wie \t) • `;`: Semikolon (Standard)
`rowName`	Elementname im ausgegebenen XML-Dokument, das jede Datenzeile umschließt **Mögliche Werte:** XML-Elementname (ohne Namespace)
`excel`	Bestimmt, ob das CSV-Textdokument von *MS Excel* erzeugt wurde (kein Standard-CSV-Format) und dementsprechend konvertiert werden muss **Mögliche Werte:** • *Aktiviert*: CSV-Textdokument im *MS Excel CSV*-Format verarbeiten • *Deaktiviert*: Standard-CSV-Textdokument verarbeiten (Standard)

`useIndexedColumns`	Den Datenelementnamen die entsprechende Spaltennummer (beginnend bei 0) hinzufügen, wenn keine Spaltenbezeichnungen verfügbar sind **Mögliche Werte:** • *Aktiviert*: Spaltennummer an den Elementnamen anhängen (z. B. *<Column_0>*) • *Deaktiviert*: Alle Datenelemente erhalten den Namen `<Column>` (Standard)
`commentStart`	Präfix zur Kennzeichnung von Kommentarzeilen; zutreffende Zeilen werden ausgelassen. **Mögliche Werte:** • Beliebige Zeichenkette (jedoch keine Trenn- oder Anführungszeichen) (z. B. ******) • *(leer)*: Keine Kommentarbehandlung; Zeilen werden als Datensätze interpretiert. (Standard)
`inputEncoding`	Zeichenkodierung des Inputs **Mögliche Werte:** • Beliebige gültige Zeichenkodierung (z. B. *UTF-8*) • *(leer)*: Standard-Zeichenkodierung der Laufzeitumgebung verwenden
`url`	URL zum einzulesenden CSV-Textdokument **Mögliche Werte:** Beliebige gültige URL (z. B. *xstore://Project/Folder/File.csv*)
`encoding`	Zeichenkodierung des CSV-Textdokuments **Mögliche Werte:** • Beliebige gültige Zeichenkodierung (z. B. *UTF-8*) • *(leer)*: Standard-Zeichenkodierung der Laufzeitumgebung verwenden

Statuswerte

-1	Fehler bei der Ausführung des Adapters (Details siehe Server-Log)
0	Der Adapter gibt keine Daten (mehr) aus. Dieser Status kann bei Operation `Iterate` als Abbruchbedingung einer Schleife im Technical Process ausgewertet werden und gibt an, dass keine weiteren Datensätze verfügbar sind.
1	Der Adapter gibt ein Ergebnis aus. Dieser Status kann bei Operation `Iterate` als Schleifeneintrittsbedingung im Technical Process ausgewertet werden und gibt an, dass ein weiterer Datensatz verfügbar ist.

Input

Der Adapter erwartet eine beliebige strukturierte CSV-Textdatei (ggf. auch von *MS Excel* erzeugt) mit einem beliebigen Trennzeichen und optionalen Kommentarzeilen.

Ausgabe

Dieser Funktions-Adapter gibt ein UTF-8-kodiertes XML-Dokument mit folgender Struktur aus. Für jede verarbeitete Zeile wird ein Element erzeugt, das ggf. den Namen aus Parameter `rowName` enthält:

```
<Csv>
   <!-- Wenn Spaltenbezeichnungen übernommen werden -->
   <Elementname rowNo="Zeilennummer im Ausgangsdokument">
      <Spaltenbezeichnung>Wert</Spaltenbezeichnung>
      <!--...-->
   </Elementname>

   <!-- Wenn keine Spaltenbezeichnungen verfügbar sind -->
   <Elementname rowNo="Zeilennummer im Ausgangsdokument">
      <Column>Wert</Column>
      <!--...-->
   </Elementname>

   <!-- Wenn keine Spaltenbezeichnungen verfügbar sind
        und Parameter 'useIndexedColumns' aktiviert ist -->
   <Elementname rowNo="Zeilennummer im Ausgangsdokument">
      <Column_Spaltennummer>Wert</Column_Spaltennummer>
        <!--...-->
   </Elementname>
</Csv>
```

Beispiel

Eine CSV-Textdatei mit Kommentaren soll verarbeitet werden, die erst in der zweiten Zeile die Spaltenbeschreibungen enthält, beispielsweise:

```
**** This line is just a comment and should not be processed. ****
"METADATA","Value 1","Value 2","Value 3","Value 4","Value 5"
"CustomerID","CompanyName","ContactName","ContactTitle","Address","PostalCode"
"ALFKI","Alfreds Futterkiste","Maria Anders","Sales","Obere Str. 57","12209"
**** This line is just a comment and should not be processed. ****
"ANTON","Moreno Taquería","Antonio Moreno","Owner","Mataderos  2312","05023"
```

Hierzu wird der Adapter-Parameter `columnNamesAfterStart` aktiviert und Parameter `start` auf *1* gesetzt, weil sowohl Kommentar- als auch Spaltenbeschriftungszeile nicht mitgezählt werden. Mit dem Wert *Dataset* für Parameter `rowName` wird beispielsweise folgendes XML-Dokument erzeugt:

```
<Csv>
  <Dataset rowNo="2">
    <CustomerID>ALFKI</CustomerID>
    <CompanyName>Alfreds Futterkiste</CompanyName>
    <ContactName>Maria Anders</ContactName>
    <ContactTitle>Sales</ContactTitle>
    <Address>Obere Str. 57</Address>
    <PostalCode>12209</PostalCode>
  </Dataset>
  <Dataset rowNo="3">
    <CustomerID>ANTON</CustomerID>
    <CompanyName>Moreno Taquería</CompanyName>
    <ContactName>Antonio Moreno</ContactName>
    <ContactTitle>Owner</ContactTitle>
    <Address>Mataderos 2312</Address>
    <PostalCode>05023</PostalCode>
  </Dataset>
</Csv>
```

10.10.5.3 XML to CSV Converter

? Unknown Attachment `XML to CSV Converter`: Konvertiert eine XML-Datei in eine CSV-Datei.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *Convert*: Konvertiert die XML-Datei in eine CSV-Datei.

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.xml2csv.XML2CSVAdapter*: Hauptklasse (Standard)

`headerInOutput`	*Erforderlich*. Legt fest, ob es in der zu erzeugenden CSV-Datei eine Kopfzeile gibt. **Mögliche Werte:** `true` (Standard) / `false`
`delimiter`	*Erforderlich*. Legt das Trennzeichen der zu erzeugenden CSV-Datei fest. **Mögliche Werte:** Beliebiges Trennzeichen, z. B. **;** , **,** (Standard) oder `\t` (Tab als Trennzeichen)
`lineEnd`	*Erforderlich*. Legt das Zeichen für das Zeilenende in der zu erzeugenden CSV-Datei fest. **Mögliche Werte:** Zeichen für das Zeilenende, z. B. `\n` (Standard Unix), `\r\n` (Standard MS Windows)
`quotechar`	Legt das Anführungszeichen für die Elemente innerhalb der zu erzeugenden CSV-Datei fest. **Mögliche Werte:** • Beliebiges Anführungszeichen, z. B. " • Standard: Nicht gesetzt
`escapechar`	Legt das Zeichen fest, mit dem bestimmte Zeichen maskiert werden, die ansonsten als Steuerzeichen erkannt würden. **Mögliche Werte:** • beliebiges Zeichen, z. B. " • Standard: Nicht gesetzt
`encoding`	Legt die Zeichenkodierung für die Konvertierung fest. **Mögliche Werte:** • Angabe einer Zeichenkodierung, z. B. `UTF-8` • Standard: `UTF-8`

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation konnte nicht ausgeführt werden.

Input

Der Adapter erwartet die folgende Input-Struktur:

Input-Struktur

```
<?xml version="1.0" encoding="UTF-8"?>
<Csv>
    <Header>
        <Column name="columnName1" />
        <Column name="columnName2" />
        <Column ...
    </Header>
    <Records>
        <Record>
            <Column name="columnName1">value1</Column>
            <Column name="columnName2">value2</Column>
            <Column ...
        </Record>
        <Record>
            <Column name="columnName1">value1</Column>
            <Column name="columnName2">value2</Column>
            <Column ...
        </Record>
        <Record ...
    </Records>
</Csv>
```

Wenn der Parameter `headerInOutput` auf *`true`* gesetzt wurde, dann wird in der Ausgabe-CSV-Datei eine Kopfzeile erzeugt. Die Spaltennamen werden über das Attribut `name` innerhalb des `Header/Column`-Elements festgelegt. Es werden so viele Spalten erzeugt, wie es `name`-Attribute gibt.
Wenn der Parameter `headerInOutput` auf *`false`* gesetzt wurde, dann wird in der Ausgabe-CSV-Datei keine Kopfzeile erzeugt. Es werden so viele Spalten erzeugt, wie maximal als `Column`-Elemente innerhalb eines `Record`-Elements vorkommen.
Jeder Eintrag muss in einem `Record`-Element stehen. Wenn eine Kopfzeile definiert ist, dann werden die Werte der Spalte mit dem entsprechenden `name` zugeordnet.

Ausgabe

Der Adapter gibt ein CSV-Dokument in der Struktur aus, die über das Eingabedokument definiert wurde.

Beispiel

Beispiel Input XML to CSV Converter

```
<?xml version="1.0" encoding="UTF-8"?>
<Csv>
    <Header>
        <Column name="Monday" />
        <Column name="Tuesday" />
        <Column name="Wednesday" />
    </Header>
    <Records>
        <Record>
            <Column name="Monday">First day</Column>
            <Column name="Tuesday">Second day</Column>
        </Record>
        <Record>
            <Column name="Monday">First day</Column>
            <Column name="Tuesday">Second day</Column>
            <Column name="Wednesday">Third day</Column>
        </Record>
    </Records>
</Csv>
```

Die Adapter-Parameter wurden wie folgt gesetzt:

- `headerInOutput`: *true*
- `delimiter`: *,*
- `encoding`: *UTF-8*

Obiger Input führt zu folgendem Output:

Beispiel Output XML to CSV Converter

```
Monday,Tuesday,Wednesday
First day,Second day,
First day,Second day,Third day
```

10.10.6 Encoding Converter

`Encoding Converter`: Konvertiert die Zeichenkodierung eines XML-Dokuments. Dabei werden sowohl die Deklaration der Zeichenkodierung (XML-Prolog) als auch die eigentliche Zeichenkodierung des Inhaltes geändert.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *`Execute`*: Zeichen der Input-XML-Daten in die in Parameter `TargetEncoding` angegebene Zeichenkodierung konvertieren

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.encodingconversion.DocumentEncodingAdapter`*: Hauptklasse (Standard)
TargetEncoding	Neue Zeichenkodierung des ausgegebenen XML-Dokuments **Mögliche Werte:** Beliebige von der Java-Laufzeitumgebung unterstützte Zeichenkodierung (z.B. *`ISO-8859-1`*)

10.10.7 Hex Converter

`Hex Converter`: Konvertiert Binärdaten hexadezimal und dekodiert hexadezimal kodierte Daten.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`Encode`*: Daten hexadezimal kodieren • *`Decode`*: Hexadezimale Daten dekodieren

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.HexAdapter`*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.10.8 Image Processing

10.10.8.1 Image to Image Converter

? Unknown Attachment `Image to Image Converter`: Konvertiert und skaliert eine Bilddatei in ein anderes Bildformat. Folgende Formate können gelesen und erzeugt werden: BMP, GIF, JPEG, JPEG 2000, PNG, RAW, TIFF und WBMP. Die Bilddateien werden in einem adapterspezifischen Input-XML-Dokument angegeben. Der Adapter bietet die Möglichkeit der Angabe von Formatierungsoptionen bei der Umwandlung.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`Transformation`* : Bildformat konvertieren

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.image2image.ImageToImageAdapter`*: Hauptklasse (Standard)
width	Breite des ausgegebenen Bildes in Pixel (wird ignoriert, wenn Parameter `maxSize` gesetzt ist) **Mögliche Werte:** Beliebige ganze Zahl
height	Höhe des ausgegebenen Bildes in Pixel (wird ignoriert, wenn Parameter `maxSize` gesetzt ist) **Mögliche Werte:** Beliebige ganze Zahl
maxSize	Wenn das Bild im ursprünglichen Seitenverhältnis skaliert werden soll: Maximale Breite oder Höhe des ausgegebenen Bildes in Pixel (Parameter `width` und `height` werden ignoriert) **Mögliche Werte:** • Beliebige ganze Zahl • *0*: Ursprüngliches Seitenverhältnis *nicht* beibehalten und Parameter width und height zur Skalierung verwenden (Standard)

scaleAlgorithm	Verwendeter Skalierungs-Algorithmus **Mögliche Werte:** • *SCALE_DEFAULT*: Standard-Algorithmus verwenden • *SCALE_FAST*: Schnellen Algorithmus verwenden • *SCALE_SMOOTH*: Kantenglättungs-Algorithmus verwenden • *SCALE_REPLICATE*: Pixelwiederholungs-Algorithmus verwenden • *SCALE_AREA-AVERAGING*: Flächen-Durchschnittsberechnungs-Algorithmus verwenden
outputFormat	Ausgegebenes Bildformat **Mögliche Werte:** • *BMP*: Windows Bitmap erzeugen • *GIF*: GIF-komprimiertes Bild erzeugen • *JPEG*: JPEG-komprimiertes Bild erzeugen • *JPEG 2000*: Bild im *JPEG 2000*-Format erzeugen • *PNG*: PNG-Bild erzeugen • *RAW*: Rohdaten-Bild erzeugen • *TIFF*: Bild im TIF-Format erzeugen • *WBMP*: Bild im *Wireless Bitmap*-Format erzeugen

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.10.8.2 Multipage TIFF Creator

? Unknown Attachment `Multipage TIFF Creator`: Erstellt ein mehrseitiges TIFF-Dokument (*Multipage Tagged Image File Format*) aus Bilddateien vom Typ BMP, GIF, FlashPix, JPEG, PNG, PNM, TIFF und WBPM. Die Bilddateien werden in einem adapterspezifischen Input-XML-Dokument angegeben. Der Adapter gibt eine mehrseitige TIFF-Grafik als Binärdatenstrom mit einer maximalen Größe von 4 GB aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** `CreateMultiTiff`: Mehrseitiges TIFF-Dokument erstellen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** `de.softproject.integration.adapter.tiff.Img2Tiff`: Hauptklasse (Standard)
compression	Kompressionstyp **Mögliche Werte:** • `Deflate`: Per Deflate verlustfrei komprimieren • `JPEG`: JPEG-Komprimierung (verlustbehaftet) verwenden • `None`: Nicht komprimieren (Standard) • `Packbits`: Per PackBits-Lauflängenkodierung verlustfrei komprimieren
deflateLvl	Deflate-Level (wenn Kompressionstyp *Deflate* gewählt wurde und die *Deflate*-Komprimierung auf das gesamte TIFF-Dokument angewendet werden soll) **Mögliche Werte:** • Beliebige ganze Zahl von `1`–`9` (komprimiert bei `1` am schnellsten, bei `9` am ausführlichsten) • `0`: Keine *Deflate*-Komprimierung auf das gesamte TIFF anwenden (Standard)
resX	Horizontaler Bildschirm-/Druck-Auflösungswert **Mögliche Werte:** • Beliebige ganze Zahl (z. B. *72*) • *0*: Keinen Auflösungswert setzen

resY	Vertikaler Bildschirm-/Druck-Auflösungswert **Mögliche Werte:** • Beliebige ganze Zahl (z. B. *72*) • *0*: Keinen Auflösungswert setzen
resUnit	Einheit für die Bildschirm-/Druck-Auflösung **Mögliche Werte:** • *`No absolute unit`*: Keine absolute Einheit setzen • *`Inch`*: Zoll als Einheit verwenden (DPI bzw. PPI) • *`Centimeter`*: Zentimeter als Einheit verwenden

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Adapter erwartet eine adapterspezifische Input-XML-Struktur, in der sämtliche zu verarbeitenden Bilddateien über ein gültiges URL-Protokoll referenziert sind. Über ein optionales Attribut thumbnail_scale können Sie angeben, ob eine skalierte Version des Bilds zusätzlich als Seite in die TIFF-Datei eingefügt werden soll. Der Attributwert gibt dabei den Skalierungsfaktor an.

```
<imagechooser>
   <image url="URL zur Bilddatei" thumbnail_scale="Skalierungsfaktor" />
</imagechooser>
```

Beispiel

Die folgende Input-XML-Struktur referenziert zwei Grafiken, eines per `xstore://` aus dem X4 Repository, das zweite über eine absolute Pfadangabe mit `file:///`. Der Funktions-Adapter erzeugt eine dreiseitige TIFF-Datei, die das erste Bild unverändert enthält, auf der zweiten Seite die um 50% verkleinerte Version des ersten Bildes sowie auf der dritten Seite das zweite Bild enthält.

```
<imagechooser>
   <image url="xstore://Folder/Image.tif" thumbnail_scale="0.5" />
   <image url="file:///C:/Folder/Picture.jpg" />
</imagechooser>
```

10.10.8.3 TIFF Properties Reader

`TIFF Properties Reader`: Ermittelt die Eigenschaften einer TIFF-Grafik. Dieser Adapter erwartet eine TIFF-Grafik als Input und gibt sämtliche ermittelten TIFF-Eigenschaften (Abmessungen, Zahl der Seiten, Bittiefe, Orientierung etc.) als XML-Dokument aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *`ReadFromImage`*: Eigenschaften aus der TIFF-Grafik auslesen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.tiff.TIFFproperties`*: Hauptklasse (Standard)

10.10.9 Java Properties File Reader

`Java Properties File Reader`: Liest eine Java-Properties-Textdatei und speichert alle oder ausgewählte Properties in einer Hash-Tabelle, die während der Laufzeit des Servers für alle Technical Processes global verfügbar ist.
Jeder Property ist dabei ein eindeutiger Schlüssel zugeordnet. Der Adapter kann zudem die globale Hash-Tabelle lesen und deren Inhalt komplett oder nach einem RegEx-Filter in einem XML-Dokument ausgeben.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`ReadFromFile`*: Properties-Textdatei einlesen und Properties komplett oder mit einem RegEx-Filter in einer globaler Hash-Tabelle speichern • *`ReadFromMap`*: Globale Hash-Tabelle lesen und deren Inhalt in einem XML-Dokument ausgeben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.properties.PropertiesReader`*: Hauptklasse (Standard)

file	Pfad zur Properties-Textdatei im Repository oder absoluter Pfad im Dateisystem des *X4 Servers* **Mögliche Werte:** • Beliebiger Repository-Pfad (z.B. *%USER_COLLECTION%/Project/Folder/sample.properties*) • Beliebiger absoluter Pfad (z.B. *C:\Data\sample.properties*)
fileKey	Schlüssel, der den eingelesenen Properties vorangestellt wird, um diese eindeutig identifizierbar zu machen. So wird beispielsweise eine Property sample.property mit dem Schlüssel `x4propkey` ergänzt und als `x4propkey.sample.property` in der Hash-Tabelle gespeichert. **Mögliche Werte:** Beliebige Zeichenkette (z.B. *x4propkey*)
filterRegex	Regulärer Ausdruck, nach dem die Properties gefiltert werden sollen (nur bei Operation `ReadFromFile`); mehrere Filterausdrücke können durch Semikolon (*;*) getrennt angegeben werden **Mögliche Werte:** Beliebiger regulärer Ausdruck (z.B. *.*sample.**)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.10.10 QR Code Generator

`QR Code Generator`: Erzeugt QR-Codes und liest diese aus.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Consume`: Liest einen QR-Code aus. • `Generate`: Erzeugt einen QR-Code.

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.qrcode.QRCodeGenerator*: Hauptklasse (Standard)

`imageWidth`	*Erforderlich*. Breite des zu erzeugenden QR-Codes in Pixel. **Mögliche Werte:** Beliebige Ganzzahl (Standard: `350`) ⓘ Der erzeugte QR-Code ist immer quadratisch. Wenn Höhe und Breite nicht identisch sind, wird mit leeren Pixeln aufgefüllt.
`imageHeight`	*Erforderlich*. Höhe des zu erzeugenden QR-Codes in Pixel. **Mögliche Werte:** Beliebige Ganzzahl (Standard: `350`) ⓘ Der erzeugte QR-Code ist immer quadratisch. Wenn Höhe und Breite nicht identisch sind, wird mit leeren Pixeln aufgefüllt.
`imageFormat`	*Erforderlich*. Dateiformat der Grafikdatei für den zu erzeugenden QR-Code. **Mögliche Werte:** • `JPEG 2000` • `PNG` • `RAW` • `TIFF` • `WBMP`

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation konnte nicht ausgeführt werden.

Input

Der Adapter erwartet abhängig von der auszuführenden Operation unterschiedliche Daten als Input.

- `Consume`-Operation: Grafik mit QR-Code
- `Generate`-Operation: Reiner Text

Output:

Der Adapter liefert abhängig von der ausgeführten Operation unterschiedliche Daten als Output.

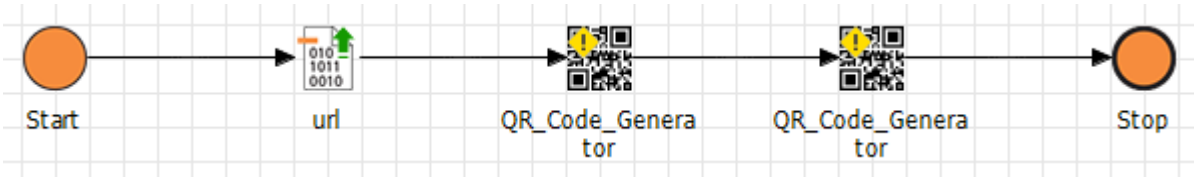
- `Consume`-Operation: Inhalt, der aus dem QR-Code ausgelesen wurde.

- `Generate`-Operation: Grafik mit QR-Code

Beispiel

Mit dem QR Code Generator können beispielsweise QR-Codes erzeugt und ausgelesen werden, die eine URL enthalten.

Der Prozess dazu sieht folgendermaßen aus:



Label	Datei	Beschreibung
url	url.txt	Textdatei, die die URL `https://softproject.de/de/` enthält.
QR_Code_Generator	Adapter QR Code Generator	Operation `Generate`
QR_Code_Generator	Adapter QR Code Generator	Operation `Consume`

Parameter des Adapters:

- `imageWidth`: 350
- `imageHeight`: 350
- `imageFormat`: PNG

10.10.11 Regular Expression

? Unknown Attachment `Regular Expression`: Durchsucht ein Input-Textdokument nach einem Muster, das in einem Regulären Ausdruck (*Regex*) angegeben ist, und gibt ein XML-Dokument aus, das für jede übereinstimmende Zeile ein XML-Element enthält.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *Grep*: Textdokument nach dem in Parameter `pattern` angegebenen Muster durchsuchen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.grep.Grep*: Hauptklasse (Standard)

pattern	Regex-Muster, nach dem gesucht wird (Regulärer Ausdruck) **Mögliche Werte:** Beliebiger Regulärer Ausdruck (z. B. `.*Test.*`)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Ausgabe

Dieser Adapter gibt ein UTF-8-kodiertes XML-Dokument mit folgender Struktur aus. Für jede übereinstimmende Zeile wird ein XML-Element `<Line/>` erzeugt, das die Zeilennummer und den Inhalt der jeweiligen Zeile enthält:

```
<Text>
   <Line lineNo="Zeilennummer">Inhalt der Zeile</Line>
</Text>
```

10.10.12 Stream Splitter

`Stream Splitter`: Zerlegt einen Datenstrom in Teile, deren Größe in Byte festgelegt werden kann. Der Adapter gibt Datenstrom-Teile mit der gewünschten Länge nacheinander aus oder gibt die Teile in aufsteigender oder absteigender Reihenfolge aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`Cut`*: Datenstrom aufteilen und jeweils einen Teil als Datenstrom ausgeben • *`Clear`*: Zähler auf Anfangswert zurücksetzen und keinen Teil ausgeben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.streamsplitter.StreamSplitter`*: Hauptklasse (Standard)
BlockSize	Größe der Blöcke in Bytes (der letzte Teil des Binärdatenstroms kann kleiner sein) **Mögliche Werte:** Beliebige ganze Zahl

SegmentCount	Adapter-Status beeinflussen (nützlich in Kombination mit einem *Condition* -Baustein, wenn nur ein bestimmter Teil des Binärdatenstroms, beispielsweise der letzte Teil, verarbeitet werden soll) **Mögliche Werte:** • *none*: Status nicht beeinflussen (Adapter erhält den Status *1*, wenn ein Teil ausgegeben wird) (Standard) • *increment*: Als Status den aktuellen Zählerwert ausgeben, beginnend bei *1* • *decrement*: Als Status den aktuellen Zählerwert ausgeben, beginnend bei der höchsten Zählernummer, und den Zählerwert um 1 dekrementieren

Statuswerte

-1 (error)	Während der Ausführung des Funktions-Adapters trat ein Fehler auf. Nähere Details finden Sie im Server-Log.
0 (successful)	Adapter wurde erfolgreich ausgeführt, gab jedoch keine Daten aus, weil keine weiteren Teile verfügbar sind (Operation `Cut`) oder weil der Adapter keine Daten verarbeitete, sondern lediglich den Iterator zurücksetzte (Operation `Clear`).
1 (successful)	Adapter wurde erfolgreich ausgeführt und gab den aktuellen Teil des Binärdatenstroms aus. Wenn der Parameter `SegmentCount` auf *increment* gesetzt wurde: Der Adapter gab den ersten Teil des Binärdatenstroms aus.
2 ... n	Nur wenn der Parameter SegmentCount auf *increment* oder *decrement* gesetzt wurde: Der Adapter gibt den zweiten oder *n* -ten Teil des Binärdatenstroms aus.

10.10.13 Structured Text

10.10.13.1 Structured Text Converter

`Structured Text Converter`: Wandelt ein XML- in ein Text-Dokument oder ein Text- in ein XML-Dokument gemäß den Regeln einer ASCII-Deskriptordatei um.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`TextToXML`*: Textdokument einlesen und in ein XML-Dokument schreiben • *`XmlToText`*: XML-Daten in ein Textdokument schreiben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.text.StructuredTextConverter*: Hauptklasse (Standard)
DescFile	Name der ASCII-Deskriptordatei, die das Textdokument in seiner Datenstruktur beschreibt. Die ASCII-Deskriptordatei sollte im Benutzerordner des *X4* Repositorys abgelegt sein (z. B. *`X4DB/1/Project/Descr.ad`*) **Mögliche Werte:** Pfad und Dateiname (z. B. *`/Project/Descr.ad`*)
encoding	Zeichensatz, in dem das Textdokument kodiert sein soll **Mögliche Werte:** • Beliebige Zeichenkette für einen Zeichensatz (z. B. *`UTF-8`*) • *(leer)*: Zeichensatz des verarbeiteten XML-Dokuments verwenden (Standard)
Strict	Bestimmt, ob der Datenstrom aus dem vorangehenden Prozess-Schritt entsprechend lang sein muss, um die vollständige Struktur zu lesen **Mögliche Werte:** • `true`: Vollständige Struktur muss gelesen werden • `false`: Datenstrom kann beliebig lang sein (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.10.13.2 COBOL Copybook Reader

`COBOL Copybook Reader`: Liest ein Textdokument im *COBOL Copybook*-Format, parst es gemäß den Regeln der ASCII-Deskriptordatei und gibt ein XML-Dokument aus.

> Zeichen, die keine gültigen XML-Zeichen sind, werden durch den *Unicode Replacement Character* `#xFFFD` ersetzt. Gültige XML-Zeichen sind folgende Unicode-Zeichen:
>
> - `#x9`
> - `#xA`
> - `#xD`
> - `#x20` – `#xD7FF`
> - `#xE000` – `#xFFFD`

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`Execute`*: Textdokument im *COBOL Copybook*-Format einlesen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.CopyBookReader`*: Hauptklasse (Standard)
DescFile	Name der ASCII-Deskriptordatei, die das Textdokument im *COBOL Copybook*-Format in seiner Datenstruktur beschreibt. Die ASCII-Deskriptordatei sollte im Benutzerordner des X4 Repositorys abgelegt sein (z.B. `X4DB/1/Project/Descr.ad`) **Mögliche Werte:** Pfad und Dateiname (z.B. *`/Project/Descr.ad`*)
Strict	Bestimmt, ob der Datenstrom aus dem vorangehenden Prozess-Schritt entsprechend lang sein muss, um die vollständige Struktur zu lesen **Mögliche Werte:** • *`ja`*: Vollständige Struktur muss gelesen werden • *`nein`*: Datenstrom kann beliebig lang sein (Standard)

10.10.13.3 COBOL Copybook Writer

`COBOL Copybook Writer`: Konvertiert ein XML-Dokument gemäß den Regeln einer ASCII-Deskriptordatei in ein Textdokument im *COBOL Copybook*-Format.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`Execute`*: XML-Daten in ein Textdokument im *COBOL Copybook*-Format schreiben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.CopyBookWriter*: Hauptklasse (Standard)
DescFile	Name der ASCII-Deskriptordatei, die das Textdokument im *COBOL Copybook*-Format in seiner Datenstruktur beschreibt. Die ASCII-Deskriptordatei sollte im Benutzerordner des X4 Repositorys abgelegt sein (z.B. `X4DB/1/Project/Descr.ad`) **Mögliche Werte:** Pfad und Dateiname (z.B. */Project/Descr.ad*)
encoding	Zeichensatz, in dem das Textdokument im *COBOL Copybook*-Format kodiert sein soll **Mögliche Werte:** • Beliebige Zeichenkette für einen Zeichensatz (z.B. *UTF-8*) • *(leer)*: Zeichensatz des verarbeiteten XML-Dokuments verwenden (Standard)
DebugMode	Debug-Modus

10.10.14 Text Replacement

? Unknown Attachment `Text Replacement`: Durchsucht ein Textdokument nach einer Zeichenkette (Groß-/Kleinschreibung wird beachtet) und gibt ein Textdokument mit derselben Zeichenkodierung aus, in dem jede gesuchte Zeichenkette durch eine neue Zeichenkette ersetzt wurde.

Falls Sie eine Zeichenkette in einem XML-Dokument ersetzen möchten, beachten Sie, dass der Adapter immer ein reines Textdokument ausgibt und es sich bei diesem somit nicht mehr zwingend um wohlgeformtes XML handelt!
Die Zeichenkodierung des Dokuments muss explizit gesetzt sein, z. B. über den *Document Properties Changer*.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *`Replace`*: Zeichenkette ersetzen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.replace.Replace`* : Hauptklasse (Standard)
searchString	Zeichenkette, nach der das Input-Textdokument durchsucht werden soll **Mögliche Werte:** Beliebige Zeichenkette
replaceString	Zeichenkette, durch die `searchString` im Ergebnis-Textdokument ersetzt werden soll **Mögliche Werte:** Beliebige Zeichenkette

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Adapter erwartet ein beliebiges Textdokument.

Ausgabe

Der Adapter gibt ein Textdokument aus, in dem der gefundene Text ersetzt wurde.

10.10.15 Transliteration Adapter

`Transliteration Adapter`: Transliteriert Text aus einem Format in ein anderes, z. B. von UTF-8 nach LATIN1 (ä → ae etc.). Der Adapter führt weder Übersetzungen durch, noch ändert er den Zeichensatz.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `GetAvailableIDs`: Führt alle verfügbaren Transliteratoren auf • `Transliterate`: Transliteriert den Text gemäß eines spezifizierten Transliterators (id) • `TransliterateFromRule`: Transliteriert den Text gemäß einer spezifizierten Regel (siehe http://userguide.icu-project.org/transforms/general/rules)

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter. transliteration.TransliterationAdapter*: Hauptklasse (Standard)
`id`	ID des verwendeten Transliterators **Mögliche Werte:** Wert, der von der Operation *`GetAvailableIDs`* ausgegeben wird
`rule`	Verwendete Regel für die Transliteratation **Mögliche Werte:** Beliebige Regel spezifiziert unter http://userguide.icu-project.org/transforms/general/rules

Statuswerte

`1`	Der Adapter wurde erfolgreich ausgeführt
`-1`	Es kam zu einem Fehler während der Adapter-Ausführung

Input

Der Adapter erwartet ein Text- oder XML-Dokument als Input.

Ausgabe

Der Adapter gibt als Ergebnis das transliterierte Text- oder XML-Dokument aus.

10.11 Geo Location

- Bing Maps
- Geo Calculation
- Google Maps

10.11.1 Bing Maps

`Bing Maps`: Greift auf *Bing Maps* über dessen Webservice-Schnittstelle in Version 1 zu und ermittelt die geografischen Informationen zu einer Adresse. Zudem kann eine Route inklusive Wegpunkten berechnet und eine URL zu einem *Bing Maps* Kartenausschnitt erzeugt werden. Der Adapter gibt seine Ergebnisse wahlweise als XML- oder JSON-Dokument aus.

Um die *Bing Maps API* nutzen zu können, wird ein *Bing Maps API Key* benötigt.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`Geocode`*: Zu einer Adresse sämtliche Geoinformationen (Anschrift, Bezirk/Land, Nation, Koordinaten etc.) ermitteln • *`ReverseGeocode`*: Zu einer Dezimalkoordinate sämtliche Geoinformationen (Anschrift, Bezirk/Land, Nation, Koordinaten etc.) ermitteln • *`GetDirections`*: Route zwischen einem Start- und Zielpunkt mit beliebig vielen Wegpunkten ermitteln • *`CreateBingURL`*: URL zu einem *Bing Maps* Webservice

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.geo.microsoft.BingMapsAdapter*: Hauptklasse (Standard)
bingMapsKey	Schlüssel zur Nutzung der *Bing Maps API*
outputFormat	Ausgabeformat für Anfragen definieren **Mögliche Werte:** • *`xml`*: XML-Dokument ausgeben • *`json`*: JSON-Dokument ausgeben

timeout	Zeitbeschränkung für den Verbindungsaufbau in Millisekunden **Mögliche Werte:** • Beliebige ganze Zahl • `0`: Keine Zeitbeschränkung • `20000`: Nach 20 Sekunden den Verbindungsaufbau abbrechen
pxHost	IP-Adresse oder Host-Name eines Proxy-Servers **Mögliche Werte:** Beliebige gültige Zeichenkette (z. B. `192.168.1.1`)
pxPort	Portnummer des Proxy-Servers **Mögliche Werte:** Beliebige ganze Zahl
pxLogin	Benutzername für den Proxy-Server **Mögliche Werte:** Beliebige Zeichenkette
pxPwd	Passwort für den Proxy-Server **Mögliche Werte:** Beliebige Zeichenkette

Statuswerte

1 (successful)	Die Anfrage war erfolgreich und ein Ergebnis wurde zurückgegeben (Entspricht dem HTTP-Status-Code 200)
0 (empty)	Die Anfrage war zwar erfolgreich, jedoch wurde kein Ergebnis zurückgegeben
-1 (failed)	Die Anfrage schlug aufgrund eines technischen Fehlers fehl
-HTTP-Status-Code	Die Anfrage wurde ausgeführt, jedoch mit einem Status-Code ungleich 200 beendet. Fehlerinterpretation entsprechend dem HTTP-Status-Code.

Weitere Informationen zu den Operationen finden Sie unter:

- Geocode
- ReverseGeocode
- GetDirections
- CreateBingURL

10.11.1.1 Operation: Geocode

`Bing Maps`**-Adapter**

Mit der Operation `Geocode` lassen sich sämtliche Geoinformationen zu einer Adresse ermitteln.

Input

Für Operation `Geocode` erwartet der Adapter eine XML-Input-Struktur nach folgendem Muster, die eine vollständige Anschrift enthält (bei unvollständigen Anschriften ist die Anfrage ggf. nicht eindeutig):

```
<Geocode>
  <Address>
     <Location><!--Location, e.g. Disneyland--></Location>
     <Street><!--Street and number--></Street> -->
     <ZIP><!--ZIP code--></ZIP>
     <City><!--City--></City>
     <Country><!--Country--></Country>
  </Address>
</Geocode>
```

Ausgabe

Für Operation `Geocode` wird ein XML- oder JSON-Dokument ausgegeben, das die Geoinformationen zur angegebenen Adresse beschreibt.

10.11.1.2 Operation: ReverseGeocode

`Bing Maps`**-Adapter**

Mit Operation `ReverseGeocode` lassen sich sämtliche Geoinformationen zu einer Dezimalkoordinate ermitteln.**Input**

Für Operation `ReverseGeocode` erwartet der Adapter eine XML-Input-Struktur, die eine Dezimalkoordinaten-Angabe enthält und nach folgendem Muster aufgebaut ist:

```
<ReverseGeocode>
    <LatLng>
        <data><!--Decimal coordinate in format NN.NNNNNNNNNNNNNN,
EE.EEEEEEEEEEEEEE--></data>
    </LatLng>
</ReverseGeocode>
```

Ausgabe

Für Operation `ReverseGeocode` wird ein XML- oder JSON-Dokument ausgegeben, das die Geoinformationen zur angegebenen Dezimalkoordinate beschreibt.

10.11.1.3 Operation: GetDirections

`Bing Maps`-Adapter

Mit Operation `GetDirections` lässt sich die Route zwischen einem Start- und Zielpunkt mit beliebig vielen Wegpunkten ermitteln.**Input**

Für Operation `GetDirections` erwartet der Adapter eine XML-Input-Struktur nach folgendem Muster, die Start-, Ziel und beliebige Wegpunkte als Dezimalkoordinaten oder Adresse enthält:

```
<Routing>
    <Origin>
        <!-- Address or GeoDecimal -->
        <Address>
            <Street><!--Street and number--></Street>
            <ZIP><!--ZIP code--></ZIP>
            <City><!--Place--></City>
            <Country><!--Country--></Country>
        </Address>
    </Origin>
    <Waypoints>
        <!-- Any number of waypoints -->
        <!-- Address or GeoDecimal -->
        <Waypoint>
            <Address>
                <Location><!--Location--></Location>
                <City><!--Place--></City>
            </Address>
        </Waypoint>
    </Waypoints>
    <Destination>
        <!-- Address or GeoDecimal -->
        <GeoDecimal>
            <Latitude><!--Latitude--></Latitude>
            <Longitude><!--Longitude--></Longitude>
        </GeoDecimal>
    </Destination>
</Routing>
```

Ausgabe

Für Operation `GetDirections` wird ein XML- oder JSON-Dokument ausgegeben, das eine für PKW berechnete Route beschreibt. Falls mehrere Routenvarianten bestehen, wird die schnellste Route ausgegeben.

10.11.1.4 Operation: CreateBingURL

`Bing Maps`**-Adapter**

Mit Operation `CreateBingleURL` lässt sich eine URL zu einer *Bing Maps* Kartenausschnitts-Grafik erzeugen.**Input**

Für Operation `CreateBingURL` erwartet der Adapter eine XML-Input-Dokument, das im Element `Path` den URL-Part der *Bing Static Maps API* sowie in verschiedenen `Param`-Elementen den zu erzeugenden Kartenauschnitt definiert. Ein Parameter mit Attribut `name="mapSize"` enthält dabei die Pixelabmessungen des zu erzeugenden Kartenauschnitts.

Beispiel-Input

```
<BingMaps>
  <Path>Imagery/Map/Road/Routes</Path>
  <Param name="mapSize">640x640</Param>

  <Param name="wp.0">
    Seattle,WA;64;1
  </Param>

  <Param name="wp.1">
    Redmond,WA;66;2
  </Param>
</BingMaps>
```

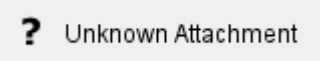
Für mehr Informationen zum Erzeugen und Formatieren von *Bing Maps* Kartenausschnitts-Grafiken, siehe Bing Static Map API unter https://msdn.microsoft.com/en-us/library/ff701724.aspx.

Ausgabe

Für Operation `CreateBingURL` wird ein XML-Dokument ausgegeben, das eine URL zu einem dem Input entsprechenden *Bing Maps*-Kartenausschnitt enthält. Das `outputFormat` wird in der URL dem Parameter `o` übergeben.

```
<HREF>
https://dev.virtualearth.net/REST/v1/Imagery/Map/Road/RoutesmapSize=640%2C640&amp;wp.
0=Seattle%2CWA%3B64%3B1&amp;wp.
1=Redmond%2CWA%3B66%3B2&amp;key=AjGQcOkb94CdwAWZxnYndqRtpQRj7_Vmy4FDZoho2zyfHfyAcsDHx
uaDLMpprFm&amp;o=xml
</HREF>
```

10.11.2 Geo Calculation



`Geo Calculation`: Konvertiert Linienzüge oder Polygone, deren Wegpunkte in den Formaten *GeoDecimal*, *GeoE5*, *GooglePoint*, *GoogleLine*, *PtvLine* oder *PtvPoint* angeben werden, in die Formate *GeoDecimal*, *GeoE5*, *GooglePoint* oder *GoogleLine*, berechnet die Länge von Linienzügen (Routenlänge) und die Kantenlängen von Polygonen, ermittelt (Kanten-) Schnittpunkte zwischen Linienzügen/Polygonen (für Gebietsgrenzberechnungen) und ermittelt Begrenzungsrahmen für ein oder mehrere Polygone/Linienzüge (z.B. zur Bestimmung eines Kartenausschnitts).

Es können für eine Linie, die ein Polygon schneidet, die darin liegenden und außerhalb liegenden Liniensegmente bestimmt werden (z.B. zur Unterscheidung von Fahrten innerhalb und außerhalb eines Gebiets). Zudem lassen sich Linienzüge/Polygone vereinfachen, sodass für eine Kartendarstellung überflüssige Wegpunkte entfernt werden.

Einschränkungen
Bei Polygon-Berechnungen, die über mehr als die Hälfte einer Erdhalbkugel gehen (mehr als 180°), können ggf. unvorhergesehene Ergebnisse auftreten. Liniensegmente dürfen nicht mehr als 170° umfassen. Falls Sie derartige Operationen durchführen möchten, kontaktieren Sie zunächst den SoftProject-Support über support@softproject.de.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`LineConversion`*: Linienzüge oder Polygone, deren Wegpunkte in den Formaten *GeoDecimal*, *GeoE5*, *GooglePoint*, *GoogleLine*, *PtvLine* oder *PtvPoint* angeben werden, in die Formate *GeoDecimal*, *GeoE5*, *GooglePoint* oder *GoogleLine* konvertieren • *`LineLength`*: Länge von Linienzügen oder die gesamte Kantenlänge von Polygonen in Kilometern berechnen • *`LineIntersection`*: (Kanten-)Schnittpunkte zwischen Polygonen oder Linienzügen ermitteln • *`BoundingBox`*: Begrenzungsrahmen (kleinstmögliches Rechteck nach Mercator-Zylinder-Projektion in der *Ebene*) für Polygone oder Linienzüge ermitteln und jeweils mit Breiten-/Höhengrad der Eckpunkte ausgeben • *`SegmentLineByPolygon`*: Für einen Linienzug, der ein Polygon schneidet, die darin liegenden und außerhalb liegenden Liniensegmente bestimmen • *`LineSimplificationPercent`*: Linienzüge/Polygone vereinfachen, sodass für eine Kartendarstellung überflüssige Wegpunkte proportional zur Gesamtstrecke entfernt werden • *`LineSimplificationKilometer`*: Linienzüge/Polygone vereinfachen, sodass für eine Kartendarstellung überflüssige Wegpunkte entfernt werden

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.geo.calc.GeoCalculationsAdapter*: Hauptklasse (Standard)
outputFormat	Geodatenformat, in dem das Ergebnis ausgegeben wird **Mögliche Werte:** • *`GeoDecimal`*: Geodezimal kodierte Koordinaten mit Breitengrad und Längengrad (Latitude/Longitude) (z.B. *`48.952/8.391`*) • *`GeoE5`*: *Geo E5*-kodierte Koordinaten mit Breitengrad und Längengrad (Latitude/Longitude) (z.B. *`4895176/839096`*) • *`GooglePoint`*: *GooglePoint*-kodierte Koordinaten (z. B. *`o{wiHozer@`*) • *`GoogleLine`*: *GoogleLine*-kodierter Linienzug, bestehend aus *GooglePoint*-Wegpunkten (z.B. *`o{wiHozer@or@ow@_yF?`*)

percent	Nur für Operation LineSimplificationPercent: Breite des Korridors, innerhalb dessen der Linienzug vereinfacht werden soll, relativ zur Gesamtstreckenlänge in Prozent geteilt durch 100 **Mögliche Werte:** • Beliebige Zahl zwischen 0 und 1 mit Punkt als Dezimaltrennzeichen (z.B. *`0.5`*) • *`0.01`*: Innerhalb eines 1 Prozent breiten Korridors alle zusätzlichen Wegpunkte entfernen (Standard)
kilometer	Nur für Operation `LineSimplificationKilometer`: Breite des Korridors, innerhalb dessen der Linienzug vereinfacht werden soll, in Kilometern **Mögliche Werte:** • Beliebige positive Zahl oder 0 mit Punkt als Dezimaltrennzeichen (z.B. *`0.5`*) • *`0.01`*: Innerhalb eines 10 Meter breiten Korridors alle zusätzlichen Wegpunkte entfernen (Standard)

Statuswerte

1 (successful)	Die Adapter-Operation wurde erfolgreich ausgeführt
0 (empty)	Bei Operation `LineIntersection`: die Zahl der Linienzüge ist nicht 2 Bei Operation `SegmentLineByPolygon`: der Input enthält kein Element Polygon, Polyline oder es fehlen beide Elemente, oder es liegt ein Fehler in der Geometrie vor (Polygon mit weniger als drei Punkte, Polyline mit weniger als zwei Punkten)
-1 (failed)	Der Adapter-Input ist kein wohlgeformtes XML-Dokument

Input

Allgemein werden für diesen Adapter Geodaten für Punkte und Linien im Input-XML-Dokument einheitlich notiert. Geodezimal kodierte Koordinaten werden dabei wie folgt angegeben:

```
<GeoDecimal>
   <Latitude><!--Breitengrad im Format NN.NNNNN--></Latitude>
   <Longitude><!--Längengrad im Format EE.EEEEE--></Longitude>
</GeoDecimal>
```

Geo E5-kodierte Koordinaten werden stets als ganzzahlige Werte (geodezimal kodierter Koordinatenwert mit 100.000 multipliziert) angegeben:

```
<GeoE5>
   <Latitude><!--Breitengrad im Format NNNNNNN--></Latitude>
   <Longitude><!--Längengrad im Format EEEEEEE--></Longitude>
</GeoE5>
```

Google Point-kodierte Koordinaten werden wie folgt angegeben:

```
<GooglePoint><!--GooglePoint-Code--></GooglePoint>
```

Google Line-kodierte Linienzüge werden wie folgt definiert:

```
<GoogleLine><!--GoogleLine-Code--></GoogleLine>
```

PTV Point-kodierte Koordinaten (in *PTV*-Mercator-Projektion) werden wie folgt angegeben:

```
<PtvPoint>
   <X><!--"Easting" als Zahl ggf. mit Komma--></X>
   <Y><!--"Northing" als Zahl ggf. mit Komma--></Y>
</PtvPoint>
```

PTV Line-kodierte Linienzüge (in PTV-Mercator-Projektion) werden wie folgt angegeben. Dabei werden 1-n Wertepaare, bestehend aus "Easting" und "Northing" aneinandergefügt, sodass eine stets gerade Anzahl an Werten hinterlegt sind. Jeder Wert wird durch ein Komma getrennt.

```
<PtvLine><!--Easting1,Northing1,Easting..,Northing...--></PtvLine>
```

10.11.2.1 Operation: Line Conversion

Mit der Operation `LineConversion` lassen sich Linienzüge oder Polygone, deren Wegpunkte in den Formaten `GeoDecimal`, `GeoE5`, `GooglePoint`, `GoogleLine`, `PtvLine` oder `PtvPoint` angegeben werden, in die Formate `GeoDecimal`, `GeoE5`, `GooglePoint` oder `>GoogleLine` konvertieren.

Input

Für die Operation `LineConversion` erwartet der Adapter eine Input-XML-Struktur nach folgendem Muster, die beliebig viele Polygone (jeweils in einem Element Polygon) oder Linienzüge (jeweils in einem Element `Polyline`) enthalten kann, deren Eck-/Wegpunkte in den Formaten `GeoDecimal`,

GeoE5, GooglePoint, GoogleLine, PtvLine oder PtvPoint innerhalb gleichnamiger Kindelemente (siehe oben) angeben werden.

Für jedes Polygon oder für jeden Linienzug (*Polyline*) können Sie beliebige Attribute definieren, die ins Ausgabedokument unverändert übernommen werden.

```
<AnyRoot>
   <!-- 0-n Linienzüge -->
   <Polyline attribut="Wert"> <!-- Beliebige Attribute -->
       <!--1-n Punktdefinitionen in den Formaten GeoDecimal, GeoE5, GooglePoint oder
PtvPoint-->
   </Polyline>

   <!-- 0-n Polygone -->
   <Polygon attribut="Wert"> <!-- Beliebige Attribute -->
       <!--1-n Punktdefinitionen in den Formaten GeoDecimal, GeoE5, GooglePoint oder
PtvPoint-->
   </Polygon>
</AnyRoot>
```

Ausgabe

Für die Operation LineConversion wird ein XML-Dokument ausgegeben, das die gleiche Struktur wie das Input-Dokument besitzt und sämtliche Attribute, die in Element Polygon oder Polyline definier t werden, unverändert enthält. Je nach gewähltem Ausgabeformat werden die Punktangaben ins Format GeoDecimal, GeoE5 oder GooglePoint konvertiert oder Linienzüge ins GoogleLine-Format konvertiert und in den entsprechenden Elementen hinterlegt.

10.11.2.2 Operation: LineLength

Mit Operation LineLength lässt sich Länge von Linienzügen oder die gesamte Kantenlänge von Polygonen in Kilometern berechnen.

Input

Für Operation LineLength erwartet der Adapter die gleiche Input-XML-Struktur wie für Operation LineConversion (siehe oben), in der Sie Polygone und/oder Linienzüge (*Polylines*) sowie ggf. zusätzliche Attribute definieren.

Ausgabe

Für Operation LineLength wird ein XML-Dokument nach folgendem Muster ausgegeben, das für jedes Element Polygon oder Polyline, das in der Input-XML-Struktur definiert wurde und ein Polygon oder einen Linienzug enthält, ein Element Length besitzt, das die Linienlänge (bei Polygonen dessen gesamte Kantenlänge) in Kilometern angibt.

Sämtliche Attribute, die im Input-XML in den Elementen Polygon oder Polyline definiert waren, werden unverändert ausgegeben.

```
<Data>
   <!-- Für jedes Polygon-/Polyline-Element -->
   <Length attribut="Wert"><!--Länge in Kilometern (mit 3 Nachkommastellen)--></
Length>
</Data>
```

10.11.2.3 Operation: LineIntersection

Mit Operation `LineIntersection` lassen sich (Kanten-)Schnittpunkte zwischen Polygonen oder Linienzügen ermitteln.

Input

Für Operation `LineIntersection` erwartet der Adapter die gleiche Input-XML-Struktur wie für Operation `LineConversion` (siehe oben), in der Sie zwei Polygone, zwei Linienzüge (*Polylines*) oder ein Polygon und einen Linienzug definieren. Zusätzliche Attribute werden jedoch nicht ins Ausgabe-XML-Dokument übernommen.

Ausgabe

Für Operation `LineIntersection` wird ein XML-Dokument ausgegeben, das in Element `Intersection` alle ermittelten (Kanten-)Schnittpunkte enthält. Je nach gewähltem Ausgabeformat werden dabei die Punktangaben in jeweils einem Kindelement `GeoDecimal`, `GeoE5` oder `GooglePoint` ausgegeben.

Alternativ kann über Ausgabeformat `GoogleLine` eine Linie in einem gleichnamigen Kindelement ausgegeben werden, die entlang dieser Schnittpunkte führt:

```
<Data>
   <Intersection>
      <!-- 0-n Schnittpunkte/-linien -->
      <Punkt-/Linienformat><!--Punkt-/Liniendefinition--></Punkt-/Linienformat>
   </Intersection>
</Data>
```

10.11.2.4 Operation: BoundingBox

Mit Operation BoundingBox lassen sich Begrenzungsrahmen (kleinstmögliches Rechteck nach Mercator-Zylinder-Projektion in der *Ebene*) für Polygone oder Linienzüge ermitteln und jeweils mit Breiten-/Höhengrad der Eckpunkte ausgeben.

Input

Für Operation `BoundingBox` erwartet der Adapter die gleiche Input-XML-Struktur wie für Operation `LineConversion` (siehe oben), in der Sie Polygone, Linienzüge (*Polylines*) sowie ggf. zusätzliche Attribute definieren.

Ausgabe

Für Operation `BoundingBox` wird ein XML-Dokument ausgegeben, das für jedes im Input-Dokument definierte Polygon oder für jeden Linienzug die Eckpunkte eines rechteckigen Begrenzungrahmens enthält. Zudem enthält es sämtliche Attribute, die im Input-XML-Dokument in Element `Polygon` oder `Polyline` definiert werden.

Die erforderlichen Eckpunkte (Breiten-/Höhengrad) zum Aufspannen des Begrenzungsrahmens werden wie folgt ausgegeben:

```
<Data>
   <!-- Für jedes Polygon-/Polyline-Element -->
   <BoundingBox attribut="Wert"> <!-- Beliebige Attribute -->
      <MinLatitude><!--Geodezimal kodierter Breitengrad von Eckpunkt 1--></
MinLatitude>
      <MaxLatitude><!--Geodezimal kodierter Breitengrad von Eckpunkt 2--></
MaxLatitude>
      <MinLongitude><!--Geodezimal kodierter Längengrad von Eckpunkt 1--></
MinLongitude>
      <MaxLongitude><!--Geodezimal kodierter Längengrad  von Eckpunkt 2--></
MaxLongitude>
   </BoundingBox>
</Data>
```

10.11.2.5 Operation: SegmentLineByPolygon

Mit Operation `SegmentLineByPolygon` lassen sich für einen Linienzug, der ein Polygon schneidet, die darin liegenden und außerhalb liegenden Liniensegmente bestimmen.

Input

Für Operation `SegmentLineByPolygon` erwartet der Adapter die gleiche Input-XML-Struktur wie für Operation `LineConversion` (siehe oben), in der Sie *genau ein* Polygon in Element `Polygon` sowie *genau einen* Linienzug (*Polyline*) in Element `Polyline` definieren. Zusätzliche Attribute werden jedoch nicht ins Ausgabe-XML-Dokument übernommen.

Ausgabe

Für Operation `SegmentLineByPolygon` wird ein XML-Dokument ausgegeben, das in den Elementen `OriginalPolygon` und `OriginalPolyline` die Punkte des ursprünglichen Polygons und Linienzugs im gewählten Ausgabeformat sowie dessen (Kanten-)Länge in Kilometern in Attribut `length` enthält.

Innerhalb von Element `FinalPolygon` und `FinalPolyline` werden die korrigierten Punkte des Polygons und Linienzugs im gewählten Ausgabeformat sowie die (Kanten-)Länge des Polygons und des Linienzugs in Attribut `length` in Kilometern ausgegeben.

Innerhalb von Element `Inside` werden für jedes Liniensegment, das innerhalb des Polygons liegt, in einem Element `Part` sämtliche Punkte ausgegeben. Je nach gewähltem Ausgabeformat werden dabei die Punktangaben in jeweils einem Kindelement `GeoDecimal`, `GeoE5` oder `GooglePoint` ausgeg

eben. Zudem werden in jeweils einem Attribut `length` die Längen der innenliegenden Liniensegmente in Kilometern ausgegeben.

Innerhalb von Element `Outside` werden für jedes Liniensegment, das außerhalb des Polygons liegt, in einem Element Part sämtliche Punkte ausgegeben. Je nach gewähltem Ausgabeformat werden dabei die Punktangaben in jeweils einem Kindelement `GeoDecimal`, `GeoE5`, `GooglePoint` oder als `GoogleLine` ausgegeben. Zudem werden in jeweils einem Attribut `length` die Längen der außenliegenden Liniensegmente in Kilometern ausgegeben.

Innerhalb von Element `Edge` werden Linien ausgegeben, die sich exakt auf der Schnittlinie befinden.

```
<Result>
   <OriginalPolyline length="Länge des ursprünglichen Linienzugs in Kilometern">
      <!-- 0-n Punkte/Linien -->
      <Punkt-/Linienformat><!--Punkt-/Liniendefinition--></Punkt-/Linienformat>
   </OriginalPolyline>
   <OriginalPolygon length="Kantenlänge des ursprünglichen Polygons in Kilometern">
      <!-- 0-n Punkte/Linien -->
      <Punkt-/Linienformat><!--Punkt-/Liniendefinition--></Punkt-/Linienformat>
   </OriginalPolygon>
   <FinalPolyline length="Länge des korrigierten Linienzugs in Kilometern">
      <!-- 0-n Punkte/Linien -->
      <Punkt-/Linienformat><!--Punkt-/Liniendefinition--></Punkt-/Linienformat>
   </FinalPolyline>
   <FinalPolygon length="Kantenlänge des korrigierten Polygons in Kilometern">
      <!-- 0-n Punkte/Linien -->
      <Punkt-/Linienformat><!--Punkt-/Liniendefinition--></Punkt-/Linienformat>
   </FinalPolygon>
   <Inside>
      <!-- 0-n Liniensegmente -->
      <Part length="Länge des innenliegenden Linienzugs in Kilometern">
         <!-- 1-n Punkte/Linien -->
         <Punkt-/Linienformat><!--Punkt-/Liniendefinition--></Punkt-/Linienformat>
      </Part>
   </Inside>
   <Outside>
      <!-- 0-n Liniensegmente -->
      <Part length="Länge des außenliegenden Linienzugs in Kilometern">
         <!-- 1-n Punkte/Linien -->
         <Punkt-/Linienformat><!--Punkt-/Liniendefinition--></Punkt-/Linienformat>
      </Part>
   </Outside>
   <Edge>
      <!-- 0-n Liniensegmente -->
      <Part length="Länge des exakt auf der Linie liegenden Liniensegments in
Kilometern">
         <!-- 1-n Punkte/Linien -->
         <Punkt-/Linienformat><!--Punkt-/Liniendefinition--></Punkt-/Linienformat>
      </Part>
   </Edge>
</Result>
```

10.11.2.6 Operation: LineSimplificationPercent / LineSimplificationKilometer

Mit den Operationen `LineSimplificationPercent` und `LineSimplificationKilometer` lassen sich gemäß dem *Douglas-Peucker-Algorithmus* Linienzüge/Polygone vereinfachen, sodass für eine Kartendarstellung überflüssige Wegpunkte innerhalb eines Korridors entfernt werden. Die Korridorbreite wird relativ zur Gesamtstrecke bzw. anhand einer absoluten Korridorbreite in Kilometern definiert.

Input

Für Operation `LineSimplificationPercent` und `LineSimplificationKilometer` erwartet der Adapter die gleiche Input-XML-Struktur wie für Operation `LineConversion` (siehe oben), in der Sie beliebige Linienzüge (*Polyline*) in jeweils einem Element `Polyline` oder beliebige Polygone in jeweils einem Element `Polygon` definieren. Zusätzliche Attribute werden ins Ausgabe-XML-Dokument übernommen.

Ausgabe

Für Operation `LineSimplificationPercent` und `LineSimplificationKilometer` wird ein XML-Dokument ausgegeben, das die gleiche Struktur wie das Input-XML-Dokument besitzt und sämtliche Attribute, die in Element `Polygon` oder `Polyline` definiert werden, unverändert enthält. Je nach Korridorbreite (definiert in Parameter `percent` bzw. `kilometer`) werden die Linienzüge/Polygone entsprechend vereinfacht. Je nach gewähltem Ausgabeformat werden die Punktangaben ins Format `GeoDecimal`, `GeoE5` oder `GooglePoint` konvertiert oder Linienzüge ins `GoogleLine`-Format konvertiert und in den entsprechenden Elementen hinterlegt

10.11.3 Google Maps

? Unknown Attachment `Google Maps`: Greift auf *Google Maps* über dessen Webservice-Schnittstelle zu und ermittelt zu einer Adresse die geografischen Informationen. Zudem kann eine Route inklusive Wegpunkten berechnet und eine URL zu einem *Google Maps*-Kartenausschnitt erzeugt werden. Der Adapter gibt seine Ergebnisse wahlweise als XML- oder JSON-Dokument aus.

> ⓘ Standardmäßig wird die kostenfreie *Google Maps API* verwendet; Um sämtliche Leistungen uneingeschränkt zu nutzen, können Sie mit diesem Adapter auch die *Google Maps API Premier* verwenden, indem Sie entsprechende Zugangsdaten in den Adapterparametern `clientID` und `privateKey` hinterlegen.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`Geocode`* : Zu einer Adresse sämtliche Geoinformationen (Anschrift, Bezirk/Land, Nation, Koordinaten etc.) ermitteln • *`ReverseGeocode`*: Zu einer Dezimalkoordinate sämtliche Geoinformationen (Anschrift, Bezirk/Land, Nation, Koordinaten etc.) ermitteln • *`GetDirections`*: Route zwischen einem Start- und Zielpunkt mit beliebig vielen Wegpunkten ermitteln • *`CreateGoogleURL`*: URL zu einer *Google Maps*-Kartenausschnitts-Grafik erzeugen (bei Nutzung der *Google Maps API Premier* zudem mit signierter URL)

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.geo.google.Google MapsAdapter`*: Hauptklasse (Standard)
clientID	Kundennummer (Client ID) zur Nutzung der *Google Maps API Premier* **Mögliche Werte:** • Beliebige gültige Zeichenkette • *`(leer)`*: Kostenlose *Google Maps API* verwenden (Standard)
privateKey	Private Key für das *Google Maps API Premier* (wenn Parameter `clientID` gesetzt ist) **Mögliche Werte:** Beliebige gültige Zeichenkette
outputFormat	*Google Maps*-Ausgabeformat für Anfragen definieren **Mögliche Werte:** • *xml*: XML-Dokument ausgeben • *json*: JSON-Dokument ausgeben

timeout	Zeitbeschränkung für den Verbindungsaufbau in Millisekunden **Mögliche Werte:** • Beliebige ganze Zahl • *0*: Keine Zeitbeschränkung • *20000*: Nach 20 Sekunden den Verbindungsaufbau abbrechen
pxHost	IP-Adresse oder Host-Name eines Proxy-Servers **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. `192.168.1.1`)
pxPort	Portnummer des Proxy-Servers **Mögliche Werte:** Beliebige ganze Zahl
pxLogin	Benutzername für den Proxy-Server **Mögliche Werte:** Beliebige Zeichenkette
pxPwd	Passwort für den Proxy-Server **Mögliche Werte:** Beliebige Zeichenkette

Statuswerte

1 (successful)	Die Anfrage war erfolgreich und ein Ergebnis wurde zurückgegeben
0 (empty)	Die Anfrage war zwar erfolgreich, jedoch wurde kein Ergebnis zurückgegeben
-1 (failed)	Die Anfrage schlug aufgrund eines technischen Fehlers fehl

Weitere Informationen zu den Operationen finden Sie unter:

- Geocode
- ReverseGeocode
- GetDirections
- CreateGoogleURL

10.11.3.1 Operation: Geocode

`Google Maps`**-Adapter**

Mit der Operation `Geocode` lassen sich zu einer Adresse sämtliche Geoinformationen (Anschrift, Bezirk/Land, Nation, Koordinaten etc.) ermitteln.

Input

Für Operation `Geocode` erwartet der Adapter eine XML-Input-Struktur nach folgendem Muster, die eine vollständige Anschrift enthält (bei unvollständigen Anschriften ist die Anfrage ggf. nicht eindeutig):

```
<Geocode>
   <Address>
      <Street><!--Straße und Hausnummer--></Street>
      <ZIP><!--Postleitzahl--></ZIP>
      <City><!--Ort--></City>
      <Country><!--Land--></Country>
   </Address>
</Geocode>
```

Ausgabe

Für Operation Geocode wird ein XML- oder JSON-Dokument ausgegeben, das die Geoinformationen (Anschrift, Bezirk/Land, Nation, Koordinaten etc.) zur angegebenen Adresse gemäß http://code.google.com/intl/en/apis/maps/documentation/geocoding/#XML beschreibt.

10.11.3.2 Operation: ReverseGeocode

`Google Maps`**-Adapter**

Mit Operation `ReverseGeocode` lassen sich zu einer Dezimalkoordinate sämtliche Geoinformationen (Anschrift, Bezirk/Land, Nation, Koordinaten etc.) ermitteln.

Input

Für Operation `ReverseGeocode` erwartet der Adapter eine XML-Input-Struktur nach folgendem Muster, die eine Dezimalkoordinatenangabe enthält:

```
<ReverseGeocode>
   <LatLng>
      <data><!--Dezimalkoordinate im Format NN.NNNNNNN, EE.EEEEEEE--></data>
   </LatLng>
</ReverseGeocode>
```

Ausgabe

Für Operation `ReverseGeocode` wird ein XML- oder JSON-Dokument ausgegeben, das die Geoinformationen (Anschrift, Bezirk/Land, Nation, Koordinaten etc.) zur angegebenen Dezimalkoordinate gemäß http://code.google.com/intl/en/apis/maps/documentation/geocoding/#XML beschreibt.

10.11.3.3 Operation: GetDirections

`Google Maps`-Adapter

Mit Operation `GetDirections` lässt sich die Route zwischen einem Start- und Zielpunkt mit beliebig vielen Wegpunkten ermitteln.

Input

Für Operation `GetDirections` erwartet der Adapter eine XML-Input-Struktur nach folgendem Muster, die Start-, Ziel und beliebige Wegpunkte als Dezimalkoordinaten, Adresse oder im GooglePoint-Format enthält:

```
<GoogleRouting xmlns:exslt="http://exslt.org/common">
   <Origin>
      <!--Dezimalkoordinate, Adresse/POI oder GooglePoint-->
   </Origin>
   <Waypoints>
      <!-- Any number of waypoints -->
      <Waypoint>
         <!--Dezimalkoordinate, Adresse/POI oder GooglePoint-->
      </Waypoint>
   </Waypoints>
   <Destination>
      <!--Dezimalkoordinate, Adresse/POI oder GooglePoint-->
   </Destination>
</GoogleRouting>
```

Adressen werden in folgender XML-Struktur angegeben:

```
   <Address>
      <Street><!--Straße und Hausnummer--></Street>
      <ZIP><!--Postleitzahl--></ZIP>
      <City><!--Ort--></City>
      <Country><!--Land--></Country>
   </Address>
```

GooglePoints werden in folgendem XML-Element angegeben:

```
    <Address>
     <Location><!--Point of Interest--></Location>
     <City><!--Ort</City>
   </Address>
```

GooglePoints werden in folgendem XML-Element angegeben:

```
    <GooglePoint><!--GooglePoint-Code--></GooglePoint>
```

Dezimalkoordinaten werden in folgender XML-Struktur angegeben:

```
<GeoDecimal>
   <Latitude><!--NN.NNNNNNN--></Latitude>
   <Longitude><!--EE.EEEEEEE--></Longitude>
</GeoDecimal>
```

Ausgabe

Für Operation `GetDirections` wird ein XML- oder JSON-Dokument ausgegeben, das eine für PKW berechnete Route gemäß http://code.google.com/intl/en/apis/maps/documentation/directions/#XML beschreibt. Falls mehrere Routenvarianten bestehen, wird die schnellste Route ausgegeben.

10.11.3.4 Operation: CreateGoogleURL

`Google Maps`**-Adapter**

Mit Operation `CreateGoogleURL` lässt sich eine URL zu einer *Google Maps*-Kartenausschnitts-Grafik erzeugen (bei Nutzung der *Google Maps API Premier* zudem mit signierter URL).

Input

Für Operation `CreateGoogleURL` erwartet der Adapter eine XML-Input-Struktur nach folgendem Muster, die in Element `Path` den URL-Part des *Google Static Maps APIs* (siehe http://code.google.com/intl/de-DE/apis/maps/documentation/staticmaps/) sowie in verschiedenen `Param`-Elementen den zu erzeugenden Kartenauschnitt definiert. Ein Parameter mit Attribut `name="size"` enthält dabei die Pixelabmessungen des zu erzeugenden Kartenauschnitts.

Jedes `Param`-Element mit Attribut `name="markers"` definiert eine Stecknadelmarkierung mit Farbe (z. B. `color:blue`) und Beschriftung (ein Großbuchstabe, z. B. `label:X`).

Ein `Param`-Element mit Attribut `name="path"` definiert, jeweils durch | getrennt, Pfad-Eigenschaften (z. B. Farbe `color:blue`) sowie eine mit dem Präfix `enc:` versehene Route, bestehend aus *GoogleLine*-kodierten Wegpunkten:

```
<GoogleMaps>
  <Path><!--/maps/api/staticmap--></Path>
  <Param name="size"><!--640x640--></Param>

  <!-- für jede Stecknadel-Markierung -->
  <Param name="markers">
    <!--color:Farbe|label:Großbuchstabe|Latitude,Longitude-->
  </Param>

  <!-- für jeden Pfad; Parameter getrennt durch "|" -->
  <Param name="path">
      <!--Parametername:Parameterwert|enc:GoogleLine-kodierter Pfad-->
  </Param>
</GoogleMaps>
```

Ausgabe

Für Operation `CreateGoogleURL` wird ein XML-Dokument ausgegeben, das eine URL zu einem dem Input entsprechenden *Google Maps*-Kartenausschnitt enthält. Das `outputFormat` hat keine Auswirkung auf die Zusammensetzung der URL.

```
<HREF>https://maps.google.de?
path=&size=&sensor=false&markers=48.952663786788165%2C+8.390965795490331&signature=9B
CZz3mKiXcWJ1e6jl4puG1XUhI=</HREF>
```

10.12 Insurance

- Audatex DV90 Converter
- GDV
- SZF Reader

10.12.1 Audatex DV90 Converter

`Audatex DV90 Converter`: Liest ein *Audatex DV90*-Kalkulationsdokument ein und gibt dessen Daten als XML-Dokument aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation dieser Native Adapter durchführt **Mögliche Werte:** *`ConvertToXml`*: *Audatex DV90*-Textdokument einlesen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.nativeadapter.NativeAdapter*: Hauptklasse (Standard)

Statuswerte

`1 (successful)`	Das eingelesene *DV90*-Kalkulationsdokument wurde erfolgreich konvertiert.
`-1 (failed)`	Die Konvertierung ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.12.2 GDV

10.12.2.1 GDV Message Plausibility Checker

`GDV Message Plausibility Checker`: Prüft eine GDV-Nachricht auf Plausibilität gemäß den Regeln einer Konfigurationsdatei

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *Execute*: GDV-Nachricht auf Plausibilität prüfen, gemäß den Regeln einer in Parameter `iniFile` festgelegten Konfigurationsdatei

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.gdvplausi.GDVPlausiPrueferWrapper*: Hauptklasse (Standard)
iniFile	Pfad zur Konfigurationsdatei (`.ini`-Format) **Mögliche Werte:** Pfad und Dateiname (z.B. */Project/GDV.ini*)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.

`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.12.2.2 GDV Message Reader

`GDV Message Reader`: Liest eine GDV-Nachricht ein, parst sie gemäß den Regeln einer Deskriptordatei und gibt ein XML-Dokument mit UTF-8-Zeichenkodierung aus. Unterstützt werden die Nachrichtentypen SN 2000 / 2003, VU/VM 2009 / 2007 / 2005 / 2002 / 1998.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *`Execute`*: GDV-Nachricht einlesen und in XML konvertieren, gemäß den Regeln einer in Parameter `descriptorFile` festgelegten Deskriptor-Datei

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.ASCIIGDV`*: Hauptklasse (Standard)
DescFile	Deskriptordatei, die die GDV-Nachricht in ihrer Datenstruktur beschreibt, siehe *Handbuch X4 Designer – ASCII-Deskriptor*. Die Deskriptordatei sollte im Benutzerordner des X4 Repositorys abgelegt sein (z.B. *`X4DB/1/Project/Descr.ad`*) **Mögliche Werte:** Pfad und Dateiname (z.B. *`/Project/Descr.ad`*)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.12.2.3 GDV Message Writer

`GDV Message Writer`: Konvertiert ein XML-Dokument gemäß den Regeln einer Deskriptordatei in eine GDV-Nachricht.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *Execute*: GDV-Nachricht aus einem XML-Dokument generieren, gemäß den Regeln einer in Parameter DescFile festgelegten Deskriptor-Datei

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.ASCIIGDV*: Hauptklasse (Standard)
DescFile	Name der Deskriptordatei, die die GDV-Nachricht in ihrer Datenstruktur beschreibt, siehe *Handbuch X4 Designer – ASCII-Deskriptor*. Die Deskriptordatei sollte im Benutzerordner des X4 Repositorys abgelegt sein (z.B. *X4DB/1/Project/Descr.ad*) **Mögliche Werte:** Pfad und Dateiname (z.B. */Project/Descr.ad*)

Statuswerte

1	Die Operation wurde erfolgreich ausgeführt.
-1	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.12.2.4 GDV-VU

GDV-VU: Mit dem GDV VU Adapter können Daten im GDV Textformat gelesen und konvertiert werden.

Es stehen zwei Möglichkeiten zur Verfügung.

- Konvertierung von Dateien im GDV-ASCII-Format in XML-Dateien
- Konvertierung von XML-Dateien in Dateien im GDV-ASCII-Format

Zusätzlich können mit dem Adapter die Datumsfelder und die dezimalen Zahlenfelder in der GDV-Datei angepasst werden.

Datumsfelder

GDV-Datumsfelder können 8, 6, 4 oder 2 Zeichen lang sein und werden vom Adapter wie folgt in ein ISO-Datumformat konvertiert

	GDV Format	ISO-Format	Beispiel		Hinweis

8	12102022	TT-MM-JJJJ	12102022	12-01-2022	Auch unvollständige Datumsangaben wie 00002008 werden in das entsprechende Format konvertiert 00-00-2008
6	092022	MM-JJJJ	092022	09-2022	
4	1022	MM-JJ	1022	10-22022	Auch unvollständige Datumsangaben wie 0000 werden in das entsprechende Format konvertiert 00-00
2	14	TT	14	14	

⚠ Bei der Konvertierung von XML ins GDV-Format werden die Daten im ISO-Format an die Länge des GDV-Feldes angepasst. Ist das Datum im ISO-Format länger als das GDV-Feld, dann wird das Datum gekürzt. Wird zum Beispiel ein ISO-Datum 2022-03-24 in ein GDV-Feld mit einer Länge von 4 Zeichen konvertiert, wird ein Datum 0322 eingefügt und die Tage des Originalformats ignoriert.

Dezimale Zahlenfelder

Dezimale GDV-Zahlenfelder bestehen aus der Länge der Gesamtzahl und der Anzahl der Dezimalstellen. Dies heißt zum Beispiel, dass bei der Konvertierung ins XML-Format ein Zahlenfeld mit der Länge „4“ und zwei Nachkommastellen zu 54,34 wird.

Anzahl der Dezimalstellen	GDV Inhalt	Konvertierter Wert
4, 3	1000	1.000
6, 5	001000	0.01000
4, 2	1200	12.00
1, 0	1	1
2, 0	01	1
4, 0	1234	1234
7, 7	1234567	0.1234567

⚠ Bei der Konvertierung von XML ins GDV-Format werden die Zahlen an die Länge des GDV-Feldes angepasst. Ist ein Feld im XML-Format kürzer als die für das GDV-Feld definierte Länge, dann werden die Zahlen mit vorangestellten Leerzeichen ergänzt.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`GDV to XML`*: Konvertiert die GDV Daten in ein XML-Format. • *`XML to GDV`*: Konvertiert die XML Daten in ein GDV-Format. • *`Get supported sets`*: Zeigt alle verfügbaren "Sets" des ausgewählten GDV-Formats an.

Parameter

`formattedXmlValues`	Definiert die Formatierung der Datums- und Zahlenfelder beim Import/ Export **Mögliche Werte** • Active: Alle GDV Nummern und Datumfelder werden wie beschrieben angepasst. GDV_Datum wird in ein ISO-Datum umgewandelt. GDV-NumFeld wird in eine Integer- oder Dezimalzahl wie im Datentyp definiert umgewandelt. • Inactive: Daten im Zieldokument werden wie im Ausgangsdokument übernommen, das heißt, dass die Daten im XML-Dokument bereits im GDV-Format sein müssen.
`trimXmlValues`	Definiert, ob Leerstellen vor oder nach dem Ausgabewert gelöscht werden. Welche Leerzeichen gelöscht werden hängt vom „Alignment" ab. Dieses gilt nicht, wenn eine XML-Datei importiert wird. **Mögliche Werte:** • Active: Leerstellen bei den Ausgabewerten werden gelöscht. Das Löschen der Leerstellen ist abhängig von der eingestellten Ausrichtung (Alignment) – dies heißt, dass nur solche Leerstellen gelöscht werden, die auf der nicht definierten Seite (unaligned) stehen. • Inactive: Die Leerzeichen werden in der Ausgabedatei belassen.
`createHeaderFooter`	Definiert, ob in der Ausgabedatei eine Kopf – und Fußzeile mit den Werten <Vorsatz> und <Nachsatz> angelegt wird. **Mögliche Werte:** • Active: "Vorsatz" und "Nachsatz" wird in der XML oder GDV Datei angelegt • Inactive: Die Elemente "Vorsatz" und "Nachsatz" werden nicht in der XML oder GDV-Datei angelegt.

gdvVersion	Definiert mit welcher GDV-Version die Daten importiert/exportiert werden. Zeigt die gewählte GDV-Version an **Mögliche Werte:** • 2009 – GDV-Version 2009 • 2013 – GDV-Version 2013 • 2015 – GDV-Version 2015 • 2018 – GDV-Version 2018

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Ausgabe

Abhängig von der gewählten Adapter-Operation wird ein XML-Dokument oder ein GDV-Dokument ausgegeben.

Bei der Operation `XML to GDV` wird bei korrekten Eingabedaten eine GDV-Datei erzeugt.

Wenn die Eingabedaten inkorrekt sind, wird eine XML-Datei mit Fehlern ausgegeben.

10.12.3 SZF Reader

? Unknown Attachment `SZF Reader`: Liest ein *SilverDAT II*-Kalkulationsdokument ein und gibt die Daten als XML-Dokument aus.
Benennung des Adapters bis Version 5.5.4: *SZF Converter*

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *Read*: *SilverDAT II*-Textdokument einlesen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.szfreader.SzfReader*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.13 Internet of Things

- IEC 61850 Client
- Modbus TCP Connector
- MQTT Publisher/Subscriber
- OPC Client Connector
- OPC UA Publisher/Subscriber
- SIMATIC S7 Connector

10.13.1 IEC 61850 Client

`IEC 61850 Client`: Verbindet sich zu einem MMS-Server und liest die Geräte- und Knoteninformationen aus.

Der Connector stellt damit eine Basis-Schnittstelle für den weltweit etablierten Kommunikationsstandard IEC 61850 für die Automatisierung von Schaltanlagen zur Verfügung.
Er agiert als Client und kann Nachrichten als klassische Client-Server-Kommunikation empfangen.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`RequestServerNodes`*: Nach bestimmten Geräten und/oder Knoten suchen und die Werte auslesen • *`RequestServerModel`*: Beim MMS-Server das Geräte- und Knotenverzeichnis abfragen und als XML ausgeben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.iec.IEC61850* : Hauptklasse (Standard)
address	IP-Adresse oder Host-Name des MMS-Servers **Mögliche Werte:** Beliebige gültige Zeichenkette (z. B. *`192.168.1.1`*)
port	Portnummer des MMS-Servers **Mögliche Werte:** Beliebige ganze Zahl
authenticationParameter	Optionaler Anmeldungsparameter zur Anmeldung am MMS-Server

responseTimeout	Zeitbeschränkung für die Server-Anfrage in Millisekunden **Mögliche Werte:** • Beliebige ganze Zahl • *0*: Keine Zeitbeschränkung • *20000*: Nach 20 Sekunden die Server-Anfrage abbrechen
messageFragmentTimeout	Zeitbeschränkung zwischen zwei eingehenden Datenpaketen in Millisekunden **Mögliche Werte:** • Beliebige ganze Zahl • *0*: Keine Zeitbeschränkung • *20000*: Nach 20 Sekunden die Verbindung abbrechen
maxTPDU	Setzt die maximale TPDU (Transfer Protocol Data Unit) **Mögliche Werte:** Beliebige ganze Zahl
maxMMSPDU	Setzt die maximale MMS PDU **Mögliche Werte:** Beliebige ganze Zahl zwischen 64 und 65000

Statuswerte

1 (successful)	Die Anfrage war erfolgreich und ein Ergebnis wurde zurückgegeben
-1 (failed)	Die Anfrage schlug aufgrund eines technischen Fehlers fehl

Input

Operation RequestServerModel

Die Operation `RequestServerModel` benötigt kein Input-XML.

Operation RequestServerNodes

Die Operation `RequestServerNodes` erwarten ein Input-XML, das wie folgt aufgebaut ist:

```
<RootElement>
    <!-- @Fc ist optional; @ObjectReference ist erforderlich -->
    <Node ObjectReference="ied1lDevice1/LPHD1"/>        <!-- Liefert alle Knoten und
Geräte innerhalb von ied1lDevice1/LPHD1 -->
    <Node ObjectReference="ied1lDevice1/LPHD1" Fc="MX"/><!-- Liefert alle Knoten und
Geräte innerhalb von ied1lDevice1/LPHD1 mit dem Function Code MX  -->
</RootElement>
```

Ausgabe

Operation RequestServerModel

Die Operation `RequestServerModel` gibt ein XML-Dokument aus, das wie folgt aufgebaut ist:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServerModel>
    <LogicalDevice Name="ied1lDevice1" ObjectReference="ied1lDevice1">
        <LogicalNode Name="LLN0" ObjectReference="ied1lDevice1/LLN0">
            <FcDataObject Name="Mod" ObjectReference="ied1lDevice1/LLN0.Mod"
                Fc="ST">
                <BdaQuality Name="q" ObjectReference="ied1lDevice1/LLN0.Mod.q"
                    Fc="ST" />
                <BdaTimestamp Name="t" ObjectReference="ied1lDevice1/LLN0.Mod.t"
                    Fc="ST" />
            </FcDataObject>
            <FcDataObject Name="Beh" ObjectReference="ied1lDevice1/LLN0.Beh"
                Fc="ST">
                <BdaInt32 Name="stVal" ObjectReference="ied1lDevice1/LLN0.Beh.stVal"
                    Fc="ST" />
                <BdaQuality Name="q" ObjectReference="ied1lDevice1/LLN0.Beh.q"
                    Fc="ST" />
                <BdaTimestamp Name="t" ObjectReference="ied1lDevice1/LLN0.Beh.t"
                    Fc="ST" />
            </FcDataObject>
        </LogicalNode>
    </LogicalDevice>
</ServerModel>
```

Operation RequestServerNodes

Die Operation `RequestServerNodes` gibt ein XML-Dokument aus. Dabei wird das Input-XML in den jeweiligen Knoten um die Antwort erweitert:

```
<?xml version="1.0" encoding="UTF-8"?>
<RootElement>
    <Node ObjectReference="ied1lDevice1/LPHD1" Fc="MX">
        <NoResult />
    </Node>
    <Node ObjectReference="ied1lDevice1/LPHD1" Fc="DC">
        <LogicalNode Name="LPHD1" ObjectReference="ied1lDevice1/LPHD1">
            <FcDataObject Name="PhyNam" ObjectReference="ied1lDevice1/LPHD1.PhyNam"
                Fc="DC">
                <BdaVisibleString Name="vendor"
                    ObjectReference="ied1lDevice1/LPHD1.PhyNam.vendor" Fc="DC" Value="
"
                    MaxLength="255" StringValue="" />
            </FcDataObject>
        </LogicalNode>
    </Node>
    <Node ObjectReference="ied1lDevice1/LPHD1">
        <LogicalNode Name="LPHD1" ObjectReference="ied1lDevice1/LPHD1">
            <FcDataObject Name="PhyHealth" ObjectReference="ied1lDevice1/
LPHD1.PhyHealth"
                Fc="ST">
                <BdaInt32 Name="stVal" ObjectReference="ied1lDevice1/
LPHD1.PhyHealth.stVal"
                    Fc="ST" Value="0" />
                <BdaQuality Name="q" ObjectReference="ied1lDevice1/LPHD1.PhyHealth.q"
                    Fc="ST" Value="AAA=" Overflow="false" Inconsistent="false"
                    Validity="GOOD" OperatorBlocked="false" Inaccurate="false"
OldData="false"
                    Oscillatory="false" BadReference="false" Failure="false"
                    Substituted="false" OutOfRange="false" Test="false" />
                <BdaTimestamp Name="t" ObjectReference="ied1lDevice1/
LPHD1.PhyHealth.t"
                    Fc="ST" Value="AAAAAAAAAAA=" Date="1970-01-01T00:00:00Z"
                    LeapSecondsKnown="false" ClockNotSynchronized="false"
TimeAccuracy="0"
                    ClockFailure="false" />
            </FcDataObject>
            <FcDataObject Name="Proxy" ObjectReference="ied1lDevice1/LPHD1.Proxy"
                Fc="ST">
                <BdaBoolean Name="stVal" ObjectReference="ied1lDevice1/
LPHD1.Proxy.stVal"
                    Fc="ST" Value="false" />
                <BdaQuality Name="q" ObjectReference="ied1lDevice1/LPHD1.Proxy.q"
                    Fc="ST" Value="AAA=" Overflow="false" Inconsistent="false"
                    Validity="GOOD" OperatorBlocked="false" Inaccurate="false"
OldData="false"
                    Oscillatory="false" BadReference="false" Failure="false"
                    Substituted="false" OutOfRange="false" Test="false" />
                <BdaTimestamp Name="t" ObjectReference="ied1lDevice1/LPHD1.Proxy.t"
                    Fc="ST" Value="AAAAAAAAAAA=" Date="1970-01-01T00:00:00Z"
                    LeapSecondsKnown="false" ClockNotSynchronized="false"
TimeAccuracy="0"
                    ClockFailure="false" />
            </FcDataObject>
```

```
            <FcDataObject Name="PhyNam" ObjectReference="ied1lDevice1/LPHD1.PhyNam"
                Fc="DC">
                <BdaVisibleString Name="vendor"
                    ObjectReference="ied1lDevice1/LPHD1.PhyNam.vendor" Fc="DC" Value="
"
                    MaxLength="255" StringValue="" />
            </FcDataObject>
        </LogicalNode>
    </Node>
</RootElement>
```

10.13.2 Modbus TCP Connector

`ModBus TCP Connector`: Verbindet sich mit Geräten über das Modbus-TCP-Protokoll.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Read`: Liest Daten vom Modbus-Server aus • `Write`: Schreibt Daten auf dem Modbus-Server

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.modbus.ModBusConnectorAdapter*: Hauptklasse (Standard)
`ip`	Server-IP
`port`	Server-Port **Mögliche Werte:** • Beliebiger Port • *502* (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt
`-1`	Es kam zu einem Fehler während der Ausführung

Input

Der Adapter erwartet eine vordefinierte XML-Struktur für die verschiedenen Operationen:

- *Operation `Read`:*

Beispiel-Input für die Operation READ

```
<?xml version="1.0" encoding="UTF-8"
<Request>
    <InputRegisters count="2" ref="0"/>
    <InputRegisters count="1" ref="3"/>
    <HoldingRegisters count="2" ref="0"/>
    <HoldingRegisters count="1" ref="3"/>
    <Discretes count="2" ref="0"/>
    <Discretes count="1" ref="3"/>
    <Coils count="2" ref="0"/>
    <Coils count="1" ref="3"/>
</Request>
```

- *Operation `Write`:*

Beispiel-Input für die Operation WRITE

```
<?xml version="1.0" encoding="UTF-8"
<Request>
    <HoldingRegister ref="11">14</HoldingRegister>
    <Coil ref="20">true</Coil>
    <Coil ref="000E1">TRUE</Coil>
</Request>
```

Ausgabe

Der Adapter gibt je nach Operation verschiedene Ergebnisse aus:

- *Operation `Read`:*

Beispiel-Ausgabe für die Operation READ

```
<?xml version="1.0" encoding="UTF-8"
<Request>
    <InputRegisters count="2" ref="0">
        <Result>
            <InputRegister ref="0">0</InputRegister>
            <InputRegister ref="1">0</InputRegister>
        </Result>
    </InputRegisters>
    <InputRegisters count="1" ref="3">
        <Result>
            <InputRegister ref="0">0</InputRegister>
        </Result>
    </InputRegisters>
    <HoldingRegisters count="2" ref="0">
        <Result>
            <HoldingRegister ref="0">0</HoldingRegister>
            <HoldingRegister ref="1">0</HoldingRegister>
        </Result>
    </HoldingRegisters>
    <HoldingRegisters count="1" ref="3">
        <Result>
            <HoldingRegister ref="0">0</HoldingRegister>
        </Result>
    </HoldingRegisters>
    <Discretes count="2" ref="0">
        Result>
            <Discrete ref="0">false</Discrete>
            <Discrete ref="1">false</Discrete>
        </Result>
    </Discretes>
    <Discretes count="1" ref="3">
        Result>
            <Discrete ref="0">false</Discrete>
        </Result>
    </Discretes>
    <Coils count="2" ref="0">
        Result>
            <Coil ref="0">false</Coil>
            <Coil ref="1">false</Coil>
        </Result>
    </Coils>
    <Coils count="1" ref="3">
        Result>
            <Coil ref="0">false</Coil>
        </Result>
    </Coils>
</Request>
```

- *Operation `Write`:*

Beispiel-Ausgabe für die Operation WRITE

```
<?xml version="1.0" encoding="UTF-8"
<Request>
    <HoldingRegister ref="11">14</HoldingRegister>
    <Coil ref="20">true</Coil>
    <Coil ref="000E1">TRUE</Coil>
</Request>
```

Beispiel-Ausgabe im Falle eines Fehlers

```
<?xml version="1.0" encoding="UTF-8"
<listErrors description="Wrong Input values for the elements of this list">
    <HoldingRegister ref="11">xx</HoldingRegister>
    <Coil ref="xxxxx">true</Coil>
    <Coil ref="000B">45</Coil>
</listErrors>
```

10.13.3 MQTT Publisher/Subscriber

- MQTT Publisher
- MQTT Subscriber

10.13.3.1 MQTT Publisher

`MQTT Publisher`: Ermöglicht das Veröffentlichen von Nachrichten in einem MQTT-Broker.

Das MQTT-Protokoll bietet eine einfache Methode zur Nachrichtenübermittlung über ein Publish/Subscribe-Modell. Damit eignet es sich besonders für die Nachrichtenübermittlung im Umfeld des Internet der Dinge, wie z. B. bei Sensoren mit niedriger Leistung oder mobilen Geräten wie Telefonen, eingebetteten Computern oder Mikrocontrollern.

ⓘ Der *MQTT Publisher* wird meist in Verbindung mit dem *MQTT Subscriber* verwendet.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Publish`: Nachricht in einem MQTT-Broker veröffentlichen

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.mqtt.MqttPublisherAdapter*: Hauptklasse (Standard)
`Alias`	Broker-Alias **Mögliche Werte:** Beliebige Zeichenkette ⓘ Der Alias kann innerhalb der Konfigurationsdatei `mqttconfig.xml` als Attribut des Elements `Broker` definiert werden.
`ClientId`	Bezeichner für den Client, der im Broker veröffentlicht **Mögliche Werte:** • Falls die Client ID in der Konfigurationsdatei `mqttconfig.xml` definiert ist: beliebige Zeichenkette • Falls die Client ID nicht in der Konfigurationsdatei `mqttconfig.xml` definiert ist: UUID ⓘ Die Client ID kann innerhalb der Konfigurationsdatei `mqttconfig.xml` für jeden Broker im Element `ClientId` definiert werden.
`clientTimeout`	Timeout für `ONLINE_ON_DEMAND`-Verbindungen (für die Operation `Publish`). Wenn `ConnectionMode` von `ONLINE_ON_DEMAND` abweicht, wird `ClientTimeout` nicht berücksichtigt. **Mögliche Werte:** Ganzzahl größer als `0`

`connectionMode`	Verbindungsmodus, der zum Erstellen des Clients ausgewählt wurde. Abhängig vom gewählten Wert bleibt die Client-Verbindung erhalten oder wird nach einer bestimmten Zeit wieder getrennt (`ClientTimeout`). **Mögliche Werte:** • `ONLINE_ON_DEMAND`: Der Parameter `ClientTimeout` wird benötigt, um den Client zu konfigurieren. • `ALWAYS_ONLINE_ON_FIRST_PUBLISH`: Der Client bleibt verbunden, sobald eine erste Veröffentlichung erfolgt. Daher wird der Parameter `ClientTimeout` nicht benötigt. ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`stateful`	Legt fest, ob der Zustand zwischen dem Client und dem Broker zustandsorientiert ist. **Mögliche Werte:** `true` / `false` ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`keystorePath`	Pfad zum Keystore. Wenn keine SSL/TLS-Konfiguration benötigt wird, um eine Verbindung zum Broker herzustellen, dann kann der Parameterwert leer sein. **Mögliche Werte:** Dateipfad ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.

`keystorePassword`	Passwort zum für `KeystorePath` angegebenen Keystore. **Mögliche Werte:** Beliebige Zeichenkette ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`Topic`	Name des Topics, in dem der Client veröffentlicht **Mögliche Werte:** Beliebige Zeichenkette ⓘ Das Topic kann innerhalb der Konfigurationsdatei `mqttconfig.xml` für jeden Broker im Attribut `filter` des Elements Element `Topic` definiert werden.
`server`	Protokoll, Host und Port für die Verbindung vom Client zum Broker. **Mögliche Werte:** • `tcp://hostname:port`: Falls keine SSL/TLS-Verbindung verwendet wird. • `ssl://hostname:port`: Falls eine SSL/TLS-Verbindung verwendet wird. ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`InputFormat`	Input-Format des Adapters **Mögliche Werte:** • `BYTES` • `BASE64_STRING` • `BASE64_XML`

`loginUsername`	Username, der verwendet wird, um eine Verbindung zu einem Broker herzustellen, der Zugangsdaten verlangt. **Mögliche Werte:** Benutzername ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`loginPassword`	Passwort, das verwendet wird, um eine Verbindung zu einem Broker herzustellen, der Zugangsdaten verlangt. **Mögliche Werte:** Passwort ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`retained`	Legt fest, ob die veröffentlichte Nachricht zu dem bestimmten Thema aufbewahrt und im Broker gespeichert wird. Jeder Client, der dieses Thema abonniert, erhält die aufbewahrte Nachricht sofort nach dem Abonnieren. **Mögliche Werte:** `true` / `false`

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Es gab einen Fehler während des Verbindungsaufbaus, der Trennung der Verbindung, der Veröffentlichung oder dem Abonnieren

Input

Für die Operation `Publish` erwartet der Adapter ein beliebiges Input-Dokument.

Ausgabe

Die Ausgabe entspricht dem Input.

Konfigurationsdatei

Um Broker, Client, SSL/TLS-Konfigurationen usw. zu definieren, kann eine Konfigurationsdatei mit einer festgelegten Struktur verwendet werden, die beim Serverstart geladen wird. Die Konfigurationsdatei `mqttconfig.xml` muss unter dem Serverpfad `X4DB/0` abgelegt werden. Wenn

Broker und Client in der Konfigurationsdatei definiert sind, dann müssen in den Adapterparametern nur die Werte für `alias` und `cliendId` angegeben werden.

Die Datei enthält folgende Informationen:

KeyStore	Pfad und Passwort des Keystores, der verwendet wird, um eine SSL-Verbindung zum Broker aufzubauen. Ist dieses optionale Element nicht angegeben, wird ein Trust-Manager angelegt, der alle Zertifikate annimmt.

Bei der Pfadangabe können sowohl absolute als auch relative Pfade zum Verzeichnis `X4DB/0` angegeben werden.

Beispiel:

Relative Pfadangabe

```
<RootElement>
    <KeyStore password="p4ssw0rd">keystore.jks</KeyStore>
</RootElement>
```

Absolute Pfadangabe

```
<RootElement>
    <KeyStore password="p4ssw0rd">C:\Path\To\Keystore\keystore.jks</
KeyStore>
</RootElement>
```

Broker	MQTT-Server
	
	Relevante Informationen werden im Attribut `alias` eingebettet. Dieses Attribut muss eindeutig sein.
	Der Broker enthält folgende Elemente:
	
	• `Server`: Broker-URL; Die Protokolle TSL und TCP werden unterstützt.
	  • *`ssl://host:port`*: Die Verbindung zum Broker findet über das TSL-Protokoll statt.
	  • ℹ Host wird durch den Host Server und Port durch den Port Server ersetzt.
	  • *`tcp://host:port`*: Die Verbindung zum Broker findet über das TCP-Protokoll (ohne SSL-Verschlüsselung) statt.
	  ℹ Host wird durch den Host Server und Port durch den Port Server ersetzt.
	• `clientId`: Eindeutige Client-ID mit folgenden Attributen:
	  • `mode`: Art der Verbindung
	  • *`ALWAYS_ONLINE_ON_STARTUP`*(Standard): Der Adapter verbindet sich beim Serverstart und hält die Verbindung die ganze Zeit aufrecht
	  • *`ALWAYS_ONLINE_ON_FIRST_PUBLISH`*: Die Verbindung wird bei der ersten Anfrage (Publish) aufgebaut und die ganze Zeit aufrechterhalten
	  • *`ONLINE_ON_DEMAND`*: Der Adapter verbindet sich, wenn noch keine Verbindung besteht und hat ein zusätzliches Attribut `timeout`, das lediglich in diesem Modus gültig ist. Das Attribut trennt die Verbindung nach der angegebenen Zeit (in Sekunden) nachdem die letzte Nachricht versendet wurde.
	  ℹ Hat `mode` einen ungültigen Wert oder ist leer, wird der Standard-Wert `ALWAYS_ONLINE_ON_STARTUP` angenommen.
	  • `timeout`: Timeout in Sekunden. Dieses Attribut ist nur für den Verbindungstyp `ONLINE_ON_DEMAND` gültig und trennt die Verbindung nach der angegebenen Zeit (in Sekunden) nachdem die letzte Nachricht versendet wurde.
	  ℹ Wenn `timeout` leer ist, wird der Standard-Wert von 20 Sekunden angenommen.

Topic	Topic das abonniert werden kann, um die veröffentlichte Nachricht zu erhalten. Es hat folgende Attribute:
	• `filter`: Name des Topic-Filters
	• `qos`: Quality of service
	  • *`AT MOST ONCE`*: Die Nachricht wird höchstens ein Mal oder kein einziges Mal zugestellt. Es erfolgt keine Benachrichtigung, wenn die Zustellung erfolgt ist.
	  • *`AT LEAST ONCE`*: Die Nachricht wird immer mindestens ein Mal zugestellt. Erhält der Sender keine Benachrichtigung, wird die Nachricht mit dem DUP-Flag erneut zugestellt, bis eine Benachrichtigung empfangen wurde.
	  • *`EXACTLY ONCE`*: Die Nachricht muss beim Sender und beim Empfänger lokal gespeichert werden, bis diese verarbeitet wurde.
	ℹ Ist das Attribut `qos` nicht angegeben, ist der Standard-Wert `AT_LEAST_ONCE`.

Beispiel-Konfiguration

```
<?xml version='1.0' encoding='UTF-8'?>
<RootElement>
    <!-- Both, relative and absolute paths to X4DB/0 are allowed -->
    <KeyStore password="password">C:/Users/MaxMustermann/Documents/MQTT/Keystore/
application.keystore</KeyStore>
    <Broker alias="Mosquitto_MQTT">
        <Server>ssl://localhost:8883</Server>
        <ClientId mode="ALWAYS_ONLINE_ON_STARTUP">C1</ClientId>
        <ClientId mode="ALWAYS_ONLINE_ON_FIRST_PUBLISH">C2</ClientId>
        <ClientId timeout="40" mode="ONLINE_ON_DEMAND">C3</ClientId>
        <ClientId timeout="20" mode="ONLINE_ON_DEMAND">C4</ClientId>
        <Topic filter="topic/action1" qos="AT_LEAST_ONCE">
            <Process url="x4db://1/X4Experience/ESB/3-Reference/1-AdapterSuites/
X4MQTTAdapter/Processes/TopicProcesses/ProcessTopic1.wrf"/>
        </Topic>
        <Topic filter="topic/action2" qos="AT_MOST_ONCE">
            <Process url="x4db://1/X4Experience/ESB/3-Reference/1-AdapterSuites/
X4MQTTAdapter/Processes/TopicProcesses/ProcessTopic2.wrf"/>
        </Topic>
        <Topic filter="topic/action1" qos="EXACTLY_ONCE">
            <Process url="x4db://1/X4Experience/ESB/3-Reference/1-AdapterSuites/
X4MQTTAdapter/Processes/TopicProcesses/ProcessTopic3.wrf"/>
        </Topic>
    </Broker>
</RootElement>
```

10.13.3.2 MQTT Subscriber

`MQTT Subscriber`: Ermöglicht das Abonnieren eines Topics in einem MQTT-Broker.

Das MQTT-Protokoll bietet eine einfache Methode zur Nachrichtenübermittlung über ein Publish/Subscribe-Modell. Damit eignet es sich besonders für die Nachrichtenübermittlung im Umfeld des Internet der Dinge, wie z. B. bei Sensoren mit niedriger Leistung oder mobilen Geräten wie Telefonen, eingebetteten Computern oder Mikrocontrollern.

Der *MQTT Subscriber* wird meist in Verbindung mit dem *MQTT Publisher* verwendet.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Subscribe`: Ermöglicht das Abonnieren eines Topics in einem MQTT-Broker. • `Unsubscribe`: Ermöglicht die Abmeldung von einem Topic in einem MQTT-Broker.

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.mqtt.MqttSubscriberAdapter*: Hauptklasse (Standard)
`Alias`	Broker-Alias **Mögliche Werte:** • Beliebige Zeichenkette ⓘ Der Alias kann innerhalb der Konfigurationsdatei `mqttconfig.xml` als Attribut des Elements `Broker` definiert werden.
`ClientId`	Bezeichner für den Client, der den Broker abonniert **Mögliche Werte:** • Falls die Client ID in der Konfigurationsdatei `mqttconfig.xml` definiert ist: beliebige Zeichenkette • Falls die Client ID nicht in der Konfigurationsdatei `mqttconfig.xml` definiert ist: UUID ⓘ Die Client ID kann innerhalb der Konfigurationsdatei `mqttconfig.xml` für jeden Broker im Element `ClientId` definiert werden.
`clientTimeout`	Timeout für ONLINE_ON_DEMAND-Verbindungen (für die Operation `Subscribe`). Wenn `ConnectionMode` von `ONLINE_ON_DEMAND` abweicht, wird `ClientTimeout` nicht berücksichtigt. **Mögliche Werte:** Ganzzahl größer als `0`

`connectionMode`	Verbindungsmodus, der zum Erstellen des Clients ausgewählt wurde. Abhängig vom gewählten Wert bleibt die Client-Verbindung erhalten oder wird nach einer bestimmten Zeit wieder getrennt (`ClientTimeout`). **Mögliche Werte:** • `ONLINE_ON_DEMAND`: Der Parameter `ClientTimeout` wird benötigt, um den Client zu konfigurieren. • `ALWAYS_ONLINE_ON_FIRST_PUBLISH`: Der Client bleibt verbunden, sobald eine erste Veröffentlichung erfolgt. Daher wird der Parameter `ClientTimeout` nicht benötigt. ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`stateful`	Legt fest, ob der Zustand zwischen dem Client und dem Broker zustandsorientiert ist. **Mögliche Werte:** `true` / `false` ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`keystorePath`	Pfad zum Keystore. Wenn keine SSL/TLS-Konfiguration benötigt wird, um eine Verbindung zum Broker herzustellen, dann kann der Parameterwert leer sein. **Mögliche Werte:** Dateipfad ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.

`keystorePassword`	Passwort zum für `KeystorePath` angegebenen Keystore. **Mögliche Werte:** Beliebige Zeichenkette ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`Topic`	Name des Topics, das der Client abonniert **Mögliche Werte:** Beliebige Zeichenkette ⓘ Das Topic kann innerhalb der Konfigurationsdatei `mqttconfig.xml` für jeden Broker im Attribut `filter` des Elements `Topic` definiert werden.
`server`	Protokoll, Host und Port für die Verbindung vom Client zum Broker. **Mögliche Werte:** • `tcp://hostname:port`: Falls keine SSL/TLS-Verbindung verwendet wird. • `ssl://hostname:port`: Falls eine SSL/TLS-Verbindung verwendet wird. ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`InputFormat`	Input-Format des Adapters **Mögliche Werte:** • `BYTES` • `BASE64_STRING` • `BASE64_XML`

`loginUsername`	Username, der verwendet wird, um eine Verbindung zu einem Broker herzustellen, der Zugangsdaten verlangt. **Mögliche Werte:** Benutzername ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`loginPassword`	Passwort, das verwendet wird, um eine Verbindung zu einem Broker herzustellen, der Zugangsdaten verlangt. **Mögliche Werte:** Passwort ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`QoS`	Quality of Service, für den Nachrichtenempfang beim Abonnement eines Topics. **Mögliche Werte:** • `AT_MOST_ONCE` • `AT_LEAST_ONCE` (Standard) • `EXACTLY_ONCE` ⓘ Dieser Parameter wird nicht benötigt, wenn `alias` und `clientId` in der Konfigurationsdatei definiert sind oder die Werte für `alias` und `clientId` bereits verwendet wurden, um eine Verbindung herzustellen.
`Process`	Pfad zum Prozess, der jedes Mal ausgeführt wird, wenn eine Nachricht des abonnierten Topics empfangen wurde **Mögliche Werte:** Prozess-Pfad im Format *`x4db://User/process.wrf`*

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Es gab einen Fehler während des Verbindungsaufbaus, der Trennung der Verbindung, der Veröffentlichung oder dem Abonnieren

Input

Der Adapter erwartet kein Input-Dokument.

Ausgabe

Die Ausgabe entspricht dem Input.

Konfigurationsdatei

Um Broker, Client, SSL/TLS-Konfigurationen usw. zu definieren, kann eine Konfigurationsdatei mit einer festgelegten Struktur verwendet werden, die beim Serverstart geladen wird. Die Konfigurationsdatei `mqttconfig.xml` muss unter dem Serverpfad `X4DB/0` abgelegt werden. Wenn Broker und Client in der Konfigurationsdatei definiert sind, dann müssen in den Adapterparametern nur die Werte für `alias` und `cliendId` angegeben werden.

Die Datei enthält folgende Informationen:

KeyStore	Pfad und Passwort des Keystores, der verwendet wird, um eine SSL-Verbindung zum Broker aufzubauen. Ist dieses optionale Element nicht angegeben, wird ein Trust-Manager angelegt, der alle Zertifikate annimmt.

ⓘ Bei der Pfadangabe können sowohl absolute als auch relative Pfade zum Verzeichnis `X4DB/0` angegeben werden.

Beispiel:

Relative Pfadangabe

```
<RootElement>
    <KeyStore password="p4ssw0rd">keystore.jks</KeyStore>
</RootElement>
```

Absolute Pfadangabe

```
<RootElement>
    <KeyStore password="p4ssw0rd">C:\Path\To\Keystore\keystore.jks</
KeyStore>
</RootElement>
```

Broker	MQTT-Server Relevante Informationen werden im Attribut `alias` eingebettet. Dieses Attribut muss eindeutig sein. Der Broker enthält folgende Elemente: • `Server`: Broker-URL; Die Protokolle TSL und TCP werden unterstützt. • *`ssl://host:port`*: Die Verbindung zum Broker findet über das TSL-Protokoll statt. • ℹ Host wird durch den Host Server und Port durch den Port Server ersetzt. • *`tcp://host:port`*: Die Verbindung zum Broker findet über das TCP-Protokoll (ohne SSL-Verschlüsselung) statt. ℹ Host wird durch den Host Server und Port durch den Port Server ersetzt. • `ClientId`: Eindeutige Client-ID mit folgenden Attributen: • `mode`: Art der Verbindung • *`ALWAYS_ONLINE_ON_STARTUP`* (Standard): Der Adapter verbindet sich beim Server-Start und hält die Verbindung die ganze Zeit aufrecht • *`ALWAYS_ONLINE_ON_FIRST_PUBLISH`*: Die Verbindung wird bei der ersten Anfrage (Publish) aufgebaut und die ganze Zeit aufrechterhalten • *`ONLINE_ON_DEMAND`*: Der Adapter verbindet sich, wenn noch keine Verbindung besteht und hat ein zusätzliches Attribut `timeout`, das lediglich in diesem Modus gültig ist. Das Attribut trennt die Verbindung nach der angegebenen Zeit (in Sekunden) nachdem die letzte Nachricht versendet wurde. ℹ Hat `mode` einen ungültigen Wert oder ist leer, wird der Standard-Wert `ALWAYS_ONLINE_ON_STARTUP` angenommen. • `timeout`: Timeout in Sekunden. Dieses Attribut ist nur für den Verbindungstyp `ONLINE_ON_DEMAND` gültig und trennt die Verbindung nach der angegebenen Zeit (in Sekunden) nachdem die letzte Nachricht versendet wurde. ℹ Wenn `timeout` leer ist, wird der Standard-Wert von 20 Sekunden angenommen.

Topic	Topic das abonniert werden kann, um die veröffentlichte Nachricht zu erhalten. Es hat folgende Attribute:
	• `filter`: Name des Topic-Filters
	• `qos`: Quality of service
	  • *`AT MOST ONCE`*: Die Nachricht wird höchstens ein Mal oder kein einziges Mal zugestellt. Es erfolgt keine Benachrichtigung, wenn die Zustellung erfolgt ist.
	  • *`AT LEAST ONCE`*: Die Nachricht wird immer mindestens ein Mal zugestellt. Erhält der Sender keine Benachrichtigung, wird die Nachricht mit dem DUP-Flag erneut zugestellt, bis eine Benachrichtigung empfangen wurde.
	  • *`EXACTLY ONCE`*: Die Nachricht muss beim Sender und beim Empfänger lokal gespeichert werden, bis diese verarbeitet wurde.
	Ist das Attribut `qos` nicht angegeben, ist der Standard-Wert `AT_LEAST_ONCE`.
	Für jedes Topic wird ein Prozess definiert, der nach Eingang der Nachricht ausgeführt werden soll. Das Element Process hat folgende Attribute:
	• `url`: Prozess-Pfad im Format
	`x4db://User/process.wrf`

Beispiel-Konfiguration

```
<?xml version='1.0' encoding='UTF-8'?>
<RootElement>
    <!-- Both, relative and absolute paths to X4DB/0 are allowed -->
    <KeyStore password="password">C:/Users/MaxMustermann/Documents/MQTT/Keystore/
application.keystore</KeyStore>
    <Broker alias="Mosquitto_MQTT">
        <Server>ssl://localhost:8883</Server>
        <ClientId mode="ALWAYS_ONLINE_ON_STARTUP">C1</ClientId>
        <ClientId mode="ALWAYS_ONLINE_ON_FIRST_PUBLISH">C2</ClientId>
        <ClientId timeout="40" mode="ONLINE_ON_DEMAND">C3</ClientId>
        <ClientId timeout="20" mode="ONLINE_ON_DEMAND">C4</ClientId>
        <Topic filter="topic/action1" qos="AT_LEAST_ONCE">
            <Process url="x4db://1/X4Experience/ESB/3-Reference/1-AdapterSuites/
X4MQTTAdapter/Processes/TopicProcesses/ProcessTopic1.wrf"/>
        </Topic>
        <Topic filter="topic/action2" qos="AT_MOST_ONCE">
            <Process url="x4db://1/X4Experience/ESB/3-Reference/1-AdapterSuites/
X4MQTTAdapter/Processes/TopicProcesses/ProcessTopic2.wrf"/>
        </Topic>
        <Topic filter="topic/action1" qos="EXACTLY_ONCE">
            <Process url="x4db://1/X4Experience/ESB/3-Reference/1-AdapterSuites/
X4MQTTAdapter/Processes/TopicProcesses/ProcessTopic3.wrf"/>
        </Topic>
    </Broker>
</RootElement>
```

10.13.4 OPC Client Connector

? Unknown Attachment `OPC Client Connector` : Stellt die Verbindung zu einem OPC-Server her und kann Werte auslesen und setzen.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** `execute`: Adapter ausführen

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** `de.softproject.integration.adapter.opc.OpcAdapter`: Hauptklasse (Standard)
`Host`	Hostname oder IP-Adresse des OPC-Servers
`Domain`	Name der Windows-Domain des OPC-Servers
`User`	Benutzername, um auf den OPC-Server zuzugreifen
`Password`	Entsprechendes Passwort des Benutzers
`Clsid`	Class-ID der OPC-Server-Software
`ProgID`	Programm-ID

Statuswerte

`1`	Die Operation wurde ohne technische Fehler ausgeführt
`-1`	Die Operation wurde mit einem technischem Fehler beendet Die Operation ist eventuell teilweise ausgeführt worden; Genauere Informationen zur Fehlerursache entnehmen Sie dem Server-Log.

Konfiguration

Im Verzeichnis `X4DB/0` muss eine Datei `opc.xml` angelegt werden, die über folgende Struktur verfügt:

```
<opc_config>
   <Connection>
       <Host></Host>
       <Domain></Domain>
       <User></User>
       <Password></Password>
       <ClsId></ClsId>
   </Connection>
   <Group name="">
       <Item name=""/>
       <Item name=""/>
   </Group>
</opc_config>
```

Für jedes Element `<Group>` und `<Item>` gilt:

- Es können beliebig viele Gruppen mit beliebig vielen Items angelegt werden.
- Eine Gruppe ist eine vom Benutzer selbst zusammengestellte Sammlung von Items, die zusammen behandelt werden sollen.
- Der Gruppenname kann selbst gewählt werden.
- Der Item-Name muss dessen Benennung auf dem OPC-Sever entsprechen.
- Lese-Operationen werden immer auf die komplette Gruppe ausgeführt.
- Ein Item darf sich in mehreren Gruppen gleichzeitig befinden.
- Beim Schreiben kann man nur auf diejenigen Items schreiben, die sich in derselben Gruppe befinden.

Input

Der Adapter erwartet spezifische Input-XML-Strukturen. Diese Input-XML-Strukturen können über XSL-Mappings in Abhängigkeit von den verarbeiteten XML-Daten dynamisch generiert werden.

```
<OPC>
   <read>
        <Group name=""/>
        <Group name=""/>
   </read>
   <write>
       <Group name="">
           <Item name="">...</Item>
           <Item name="">...</Item>
       </Group>
    </write>
    <copy>
       ...
   </copy>
</OPC>
```

Erläuterung

Für jedes Element `<read>` gilt:

- Auf dem `read`-Element können beliebige Attribute angegeben werden, die in die Ausgabe kopiert werden.
- Ein `read`-Element kann beliebig viele `Group`-Elemente enthalten.

Für jedes Element `<write>` gilt:

- Auf dem `write`-Element können beliebige Attribute angegeben werden, die in die Ausgabe kopiert werden.
- Ein `write`-Element kann beliebig viele `Group`-Elemente mit beliebig vielen `Item`-Elementen ent halten.

Für jedes Element `<copy>` gilt:

- Auf dem `copy`-Element können beliebige Attribute angegeben werden, die in die Ausgabe kopiert werden.
- Das `copy`-Element kann beliebige XML-Elemente mit beliebigem Inhalt enthalten; Diese Elemente werden in die Ausgabe kopiert.

Ausgabe

> ⓘ Für alle Kommandos werden die jeweiligen Attribute entsprechend auf die Container-Elemente kopiert.

Kommando: read

```
<OPC>
 <read>
 <Group name=''>
 <Item name=''> ... </Item>
 <Item name=''> ... </Item>
 </Group>
 </read>
</OPC>
```

Kommando: write

Das Kommando wird ausgeführt, es wird jedoch keine Ausgabe generiert:

```
<OPC>
 <write/>
</OPC>
```

Kommando: copy

```
<OPC>
 <copy> <!-- XML-Struktur aus dem Input-Dokument --> </copy>
</OPC>
```

10.13.5 OPC UA Publisher/Subscriber

- OPC UA Publisher Adapter
- OPC UA Subscriber Adapter

10.13.5.1 OPC UA Publisher Adapter

`OPC UA Publisher Adapter`: Verbindet sich mit einem OPC UA Server zum Lesen und Schreiben von Daten von/zu einer Maschine in einem Technical Process.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `GetServerConfiguration`: Ruft die Server-Konfiguration auf • `ListNodes`: Listet die verfügbaren Knoten auf • `ListNodesAttributtesId`: Listet die Attributes für jeden Knoten auf • `Read`: Liest Werte und den Status aus • `ReadHistory`: Liest historische Werte und den Status aus • `Write`: Schreibt die Attribute eines Knotens

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.opcua.OpcUaPublisherAdapter*: Hauptklasse (Standard)
`url`	Server-URL **Mögliche Werte:** Beliebige gültige URL, z. B. *`opc.tcp://SP-WS-204.softproject.local:53530/OPCUA/SimulationServer`*
`user`	Benutzername, um eine Verbindung zum Server aufzubauen **Mögliche Werte:** Beliebige Zeichenkette mit dem Benutzernamen, z. B. *`myUser`*
`password`	Passwort, um eine Verbindung zum Server aufzubauen **Mögliche Werte:** Beliebige Zeichenkette mit dem Passwort, z. B. *`myPassword`*

`securityMode`	Sicherheits-Modus und Server-Richtlinien **Mögliche Werte:** • `NONE` • `BASIC256_SIGN` • `BASIC256_SIGN_ENCRYPT` • `BASIC128RSA15_SIGN` • `BASIC128RSA15_SIGN_ENCRYPT` • `BASIC256SHA256_SIGN` • `BASIC256SHA256_SIGN_ENCRYPT`
`autoReconnect`	Angabe, ob sich der Client automatisch wieder mit dem Server verbinden soll oder nicht.
`nodeIndex`	Knoten-Index **Mögliche Werte:** Beliebige ganze Zahl mit dem Knoten-Index, z. B. *`1234`*
`nodeIdentifier`	Knoten-Bezeichner **Mögliche Werte:** Beliebige Zeichenkette mit dem Knoten-Bezeichner, z. B. *`myNode`*
`nodeIdentifierType`	Typ des Knoten-Bezeichners **Mögliche Werte:** • `String` • `Numeric`
`historyOption`	Optionen zum Abrufen der Historie eines bestimmten Knotens **Mögliche Werte:** • `AT_TIMES` • `AVERAGE` • `COUNT` • `DELTA` • `END` • `MAXIMUM` • `MAXIMUM_ACTUAL_TIME` • `MINIMUM` • `MINIMUM_ACTUAL_TIME` • `RANGE` • `RAW_DATA` • `START` • `WORST_QUALITY`
`from`	Beginn-Datum mit dem Format `yyyy-MM-dd hh:mm:ss` **Mögliche Werte:** Beliebiges Datum mit dem entsprechenden Format, z. B. *`2018-01-01 00:00:00`*

`to`	End-Datum mit dem Format `yyyy-MM-dd hh:mm:ss` **Mögliche Werte:** Beliebiges Datum mit dem entsprechenden Format, z. B. *`2018-12-31 00:00:00`*
`interval`	Intervall-Wert in Sekunden **Mögliche Werte:** Beliebige ganze Zeit mit dem Intervall in Sekunden, z. B. *`3000`*
`arrayDelimiter`	Zeichen zum Trennen der Werte innerhalb eines Arrays **Mögliche Werte:** Beliebiges Zeichen (Standard `,`)

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt
`-1`	Bei der Ausführung des Adapters kam es zu einem Fehler

Input

Für die Operationen `Read` und `Write` erwartet der Adapter ein spezifisches XML-Dokument mit folgenden Elementen:

- `nodeIndex` (Integer)
- `nodeIdentifier` (String)
- `nodeIdentifierType` (String , Numeric)
- `attributeId` (Integer)

- Verwenden Sie die Operation `ListNodesAttributtesId`, um die Knoten-Attribut-ID aufzurufen.
- Arrays geben Sie an, indem Sie die Werte in eckigen Klammern angeben, z. B. [997, 998, 999].

Beispiel-Input für die Operation READ

```
<?xml version="1.0" encoding="UTF-8"
<RootElement>
    <NodeID nodeIndex="5" nodeIdentifier="Counter1" nodeIdentifierType="String"
attributeId="13"></NodeID>
    <NodeID nodeIndex="5" nodeIdentifier="Square1" nodeIdentifierType="String"
attributeId="13"></NodeID>
    <NodeID nodeIndex="5" nodeIdentifier="Triangle1" nodeIdentifierType="String"
attributeId="13"></NodeID>
    <NodeID nodeIndex="0" nodeIdentifier="2256" nodeIdentifierType="Numeric"
attributeId="5"></NodeID>
</RootElement>
```

Beispiel-Input für die Operation WRITE ohne Arrays

```
<?xml version="1.0" encoding="UTF-8"
<RootElement>
    <NodeID nodeIndex="5" nodeIdentifier="Counter1" nodeIdentifierType="String"
attributeId="13">999</NodeID>
    <NodeID nodeIndex="5" nodeIdentifier="Square1" nodeIdentifierType="String"
attributeId="13">9636</NodeID>
    <NodeID nodeIndex="5" nodeIdentifier="Triangle1" nodeIdentifierType="String"
attributeId="13">TestData</NodeID>
    <NodeID nodeIndex="0" nodeIdentifier="2256" nodeIdentifierType="Numeric"
attributeId="5">Testing</NodeID>
</RootElement>
```

Beispiel-Input für die Operation WRITE mit Arrays

```
<?xml version="1.0" encoding="UTF-8"?>
<RootElement>
    <NodeID nodeIndex="6" nodeIdentifier="BooleanArray" nodeIdentifierType="String"
attributeId="13">[true, true, true, true, true]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="ByteArray" nodeIdentifierType="String"
attributeId="13">[1, 2, 3, 4, 5]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="ByteStringArray" nodeIdentifierType="String
" attributeId="13">[[3] 0x010203, [3] 0x040506, [3] 0x070809, [3] 0x010101]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="DateTimeArray" nodeIdentifierType="String"
attributeId="13">[2021-01-23 09:33:08.0031200 GMT, 2021-02-23 09:33:08.0031200 GMT]</
NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="DoubleArray" nodeIdentifierType="String"
attributeId="13">[1.0, 2.0, 3.0, 4.0, 5.0]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="FloatArray" nodeIdentifierType="String"
attributeId="13">[1.0, 2.0, 3.0, 4.0, 5.0]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="GUIDArrayArray" nodeIdentifierType="String"
 attributeId="13">[66c6a6fa-79a8-4a47-b0d4-6416f3836bf5, abdb9e68-
dd91-4a85-84a1-54a1620209fd]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="Int16Array" nodeIdentifierType="String"
attributeId="13">[1, 2, 3, 4, 5]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="Int32Array" nodeIdentifierType="String"
attributeId="13">[1, 2, 3, 4, 5]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="Int64Array" nodeIdentifierType="String"
attributeId="13">[1, 2, 3, 4, 5]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="LocalizedTextArray" nodeIdentifierType="Str
ing" attributeId="13">[(de) Text1, (de) Text2, (de) Text3, (de) Text4, (de) Text5,
(de) Text0]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="QualifiedNameArray" nodeIdentifierType="Str
ing" attributeId="13">[Default Binary, Default XML, Default Binary, Default XML,
Default Binary]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="SByteArray" nodeIdentifierType="String"
attributeId="13">[1, 2, 3, 4, 5]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="StringArray" nodeIdentifierType="String"
attributeId="13">[testString1, testString2, testString3, testString4, testString5]</
NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="UInt16Array" nodeIdentifierType="String"
attributeId="13">[1, 2, 3, 4, 5]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="UInt32Array" nodeIdentifierType="String"
attributeId="13">[1, 2, 3, 4, 5]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="UInt64Array" nodeIdentifierType="String"
attributeId="13">[1, 2, 3, 4, 5]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="VariantArray" nodeIdentifierType="String"
attributeId="13">[true, true, true, true, true, true]</NodeID>
    <NodeID nodeIndex="6" nodeIdentifier="XmlElementArray" nodeIdentifierType="String
" attributeId="13"><![CDATA[<?xml version="1.0" encoding="utf-8" standalone="no"?>
<testElement1/>
, <?xml version="1.0" encoding="utf-8" standalone="no"?>
<testElement2/>
, <?xml version="1.0" encoding="utf-8" standalone="no"?>
<testElement3/>
, <?xml version="1.0" encoding="utf-8" standalone="no"?>
```

```
<testElement4/>
, <?xml version="1.0" encoding="utf-8" standalone="no"?>
<testElement5/>
]]]></NodeID>
</RootElement>
```

Output

Der Adapter gibt ein spezifisches XML-Dokument mit dem Ergebnis aus:

Beispiel-Output für die Operation READ

```
<?xml version="1.0" encoding="UTF-8"
<Result>
    <NodeID statusCode="GOOD" nodeIndex="5" nodeIdentifier="Counter1"
nodeIdentifierType="String">999</NodeID>
    <NodeID statusCode="GOOD" nodeIndex="5" nodeIdentifier="Square1"
nodeIdentifierType="String">9636.0</NodeID>
    <NodeID statusCode="GOOD" nodeIndex="5" nodeIdentifier="Triangle1"
nodeIdentifierType="String">1.5999999955515</NodeID>
    <NodeID statusCode="GOOD" nodeIndex="0" nodeIdentifier="2256" nodeIdentifierType="
Numeric">The current status of the server</NodeID>
</Result>
```

Beispiel-Output für die Operation WRITE

```
<?xml version="1.0" encoding="UTF-8"
<RootElement>
    <NodeID nodeIndex="5" nodeIdentifier="Counter1" nodeIdentifierType="String">SUCCE
SS</NodeID>
    <NodeID nodeIndex="5" nodeIdentifier="Square1" nodeIdentifierType="String">SUCCES
S</NodeID>
    <NodeID nodeIndex="5" nodeIdentifier="Triangle1">XML malformed - Invalid value
format to write into the Node attribute</NodeID>
    <NodeID nodeIndex="0" nodeIdentifier="2256">Bad_NotWritable (0x803B0000) "The
access level does not allow writing to the Node</NodeID>
</RootElement>
```

10.13.5.2 OPC UA Subscriber Adapter

`OPC UA Subscriber Adapter`: Stellt eine Verbindung zu einem OPC UA Server her und abonniert oder storniert Datenänderungen oder Ereignisse.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Subscribe`: Abonniert Datenänderungen oder Ereignisse entsprechend dem Input-XML-Dokument • `UnSubscribe`: Storniert bestehende Abonnements für einen bestimmten Client

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.opcua.OpcUaSubscriberAdapter*: Hauptklasse (Standard)
`clientAlias`	Alias zur Kennzeichnung der Verbindungseinstellungen **Mögliche Werte:** Beliebige Zeichenkette mit dem Client Alias
`process`	X4DB-Pfad zu einem Technical Process, der nach jedem Abonnentenempfang ausgeführt wird **Mögliche Werte:** Beliebiger gültiger X4DB-Pfad, z. B. *`x4db://1/SampleProject/SampleProcess.wrf`*
`url`	Server-URL **Mögliche Werte:** Beliebige gültige URL, z. B. *`opc.tcp://SP-WS-204.softproject.local:53530/OPCUA/SimulationServer`*
`user`	Benutzername, um eine Verbindung zum Server aufzubauen **Mögliche Werte:** Beliebige Zeichenkette mit dem Benutzernamen, z. B. *`myUser`*
`password`	Passwort, um eine Verbindung zum Server aufzubauen **Mögliche Werte:** Beliebige Zeichenkette mit dem Passwort, z. B. *`myPassword`*

`security mode`	Sicherheits-Modus und Server-Richtlinien **Mögliche Werte:** • `NONE` • `BASIC256_SIGN` • `BASIC256_SIGN_ENCRYPT` • `BASIC128RSA15_SIGN` • `BASIC128RSA15_SIGN_ENCRYPT` • `BASIC256SHA256_SIGN` • `BASIC256SHA256_SIGN_ENCRYPT`
`autoReconnect`	Angabe, ob sich der Client automatisch wieder mit dem Server verbinden soll oder nicht.

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt
`-1`	Bei der Ausführung des Adapters kam es zu einem Fehler

Input

Operation `Subscribe`:

- *Vollständiges Beispiel mit verschiedenen Abonnement-Typen:*

Input für die Operation Subscribe

```
<?xml version="1.0" encoding="UTF-8"?>
<Subscriptions>
    <!-- Subscription to a data value change -->
    <Subscription type="DATA_CHANGE" nodeIndex="3" nodeIdentifier="Counter" nodeIdentifierType="String"></Subscription>

    <!-- Subscription to an event, specifying a condition -->
    <Subscription type="EVENT" nodeIndex="5" nodeIdentifier="MyLevel.Alarm" nodeIdentifierType="String">
        <Field name="SourceName">
            <Literal>LIKE</Literal>
            <Operand>Server</Operand>
        </Field>
    </Subscription>

    <!-- Subscription to an event, specifying more than one condition -->
    <Subscription type="EVENT" nodeIndex="5" nodeIdentifier="MyLevel.Alarm" nodeIdentifierType="String">
        <Field name="ActiveState" attributeId="13">
            <Operand>EQUALS</Operand>
            <Literal>Active</Literal>
        </Field>
        <Field name="AckedState" attributeId="13">
            <Operand>EQUALS</Operand>
            <Literal>Unacknowledged</Literal>
        </Field>
        <Field name="Comment" attributeId="13"></Field>
        <Field name="EnabledState" attributeId="13"></Field>
        <Field name="EventType" attributeId="1">
            <Operand>EQUALS</Operand>
            <Literal>i=9482</Literal>
        </Field>
        <Field name="Quality">
            <Operand>EQUALS</Operand>
            <Literal>GOOD (0x00000000) ""</Literal>
        </Field>
        <Field name="Severity">
            <Operand>LESS_THAN</Operand>
            <Literal>600</Literal>
        </Field>
        <Field name="SourceName"></Field>
        <Field name="Time"></Field>
    </Subscription>

    <!-- Configuring  -->
    <OnAfterCreate>x4db://1/TestingOPCUA/Processes/OnAfterCreate.wrf</OnAfterCreate>
    <OnAlive>x4db://1/TestingOPCUA/Processes/OnAlive.wrf</OnAlive>
</Subscriptions>
```

- *Datenwertänderung abonnieren:*

  > Um eine Datenwertänderung zu abonnieren, muss ein `Subscription`-Knoten mit folgenden Pflichtattributen angelegt werden:
  > - `type`, um die Art des Abonnements festzulegen
  > - `nodeIndex`, `nodeIdentifier`, um den OPC UA-Knoten zu identifizieren, von dem die Änderungen abonniert werden
  > - `nodeIdentifierType`, um den Datentyp des Knotenbezeichners zu spezifizieren
  >
  > Der `Subscription`-Knoten kann folgende optionale Attribute enthalten:
  > - `samplingInterval`: Wert in Millisekunden, mit dem der OPC UA Server, die Datenquelle (SPS) abtastet.
  > - `publishingInterval` Wert in Millisekunden, mit dem der OPC UA Server, die Daten an den Client (z.B. X4) sendet.

```
<?xml version="1.0" encoding="UTF-8"?>
<Subscriptions>
    <Subscription type="DATA_CHANGE" nodeIndex="3" nodeIdentifier="C
ounter" nodeIdentifierType="String" samplingInterval="1000"
publishingInterval="1000">
    </Subscription>
</Subscriptions>
```

- *Events abonnieren, ohne Felder zu spezifizieren:*

```
<?xml version="1.0" encoding="UTF-8"?>
<Subscriptions>
    <!-- We will receive an empty event -->
    <Subscription type="EVENT" nodeIndex="5" nodeIdentifier="MyLevel
.Alarm" nodeIdentifierType="String">
    </Subscription>
</Subscriptions
```

- *Events abonnieren und Felder spezifizieren:*

```
<?xml version="1.0" encoding="UTF-8"?>
<Subscriptions>
    <!-- In the Field node, we define the field we want to retrieve
on each event reception. -->
    <Subscription type="EVENT" nodeIndex="5" nodeIdentifier="MyLevel
.Alarm" nodeIdentifierType="String">
        <Field name="SourceName"></Field>
        <Field name="Time"></Field>
    </Subscription>
    <!-- Any number of Field nodes possible. -->
</Subscriptions>
```

- *Events abonnieren und Filter für die spezifizierten Felder definieren:*
  Es ist möglich, Bedingungen für die Attribute zu definieren, die bei jedem Ereignisempfang

erhalten werden. Dazu müssen entsprechend viele `Field`-Knoten angelegt werden. Jeder `Field`-Knoten enthält dabei die Elemente `Literal` und `Operand`.

Event mit Filtern abonnieren

```
<?xml version="1.0" encoding="UTF-8"?>
<Subscriptions>
    <!-- In the Field node, we define the field we want to retrieve on each event reception. Inside this node Operand and Literal are defined in order to create the condition over the field. -->
    <Subscription type="EVENT" nodeIndex="5" nodeIdentifier="MyLevel.Alarm" nodeIdentifierType="String">
    <!-- Severity less than 600 -->
    <Field name="Severity">
        <Operand>LESS_THAN</Operand>
        <Literal>600</Literal>
    </Field>
    </Subscription>
    <!-- Any number of Field nodes possible. If there is more than one Field node containing conditions, the logical operator applied between fields will be 'OR' -->
</Subscriptions>
```

Beachten Sie:

- Das *name*-Attribut innerhalb von `Field` ist erforderlich.
- `Operand` kann folgende Werte haben:
  - `LESS_THAN`
  - `LESS_THAN_OR_EQUALS`
  - `GREATER_THAN`
  - `GREATER_THAN_OR_EQUALS`
  - `EQUALS`
  - `NOT`
  - `LIKE`
- `Literal` enthält die verschiedenen literalen Werte, die innerhalb des Ausdrucks berücksichtigt werden sollen.

Operation `Unsubscribe`

Die Operation `UnSubscribe` benötigt keinen Input, da eine Stornierung des Abonnements erfolgt. Daher muss im Parameter `clientAlias` lediglich der Alias des Mandanten angegeben werden.

Output

Operation `Subscribe`

Die Ausgabestruktur hängt vom Typ der Empfangsnachricht ab.

- Wird ein Event mit Feldern empfangen (mit oder ohne Bedingungen), die im Abonnement konfiguriert wurden, ähnelt die Antwort-Struktur folgender Beispielausgabe:

Output für Event-Antworten

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Message type="EVENT">
    <Node description="ns=5;s=MyLevel.Alarm"/>
    <Fields>
        <Field fieldName="ActiveState">Active</Field>
        <Field fieldName="AckedState">Unacknowledged</Field>
        <Field fieldName="Comment">EMPTY</Field>
        <Field fieldName="EnabledState">Enabled</Field>
        <Field fieldName="EventType">i=9482</Field>
        <Field fieldName="Quality">GOOD</Field>
        <Field fieldName="Severity">500</Field>
        <Field fieldName="SourceName">MyLevel</Field>
        <Field fieldName="Time">1576741128000</Field>
    </Fields>
</Message>
```

Die Angabe von Bedingungen führt dazu, dass mehr oder weniger Events empfangen werden. Sobald jedoch ein Event empfangen wird, enthält die Antwort die Felder, die im Abonnement angegeben wurden.

- Wird ein Event ohne Feld-Spezifikationen empfangen, ähnelt die Antwort-Struktur folgender Beispielausgabe:

Output für Event-Antworten

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Message type="EVENT"/>
```

- Wird eine Datenwertänderung empfangen, ähnelt die Antwort-Struktur folgender Beispielausgabe:

Output für Datenwertänderungen

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Message type="DATA_CHANGE">
    <Node description="ns=3;s=Counter"/>
    <OldValue>
        <!-- 0 means null timestamp  -->
        <sourceTimestamp>0</sourceTimestamp>
        <!-- No value and statusCode present in 'OldValue' node
happens only just after subscribing on the first reception -->
    </OldValue>
    <NewValue>
        <sourceTimestamp>13221214634996</sourceTimestamp>
        <statusCode>GOOD</statusCode>
        <value>2312.11</value>
    </NewValue>
</Message>
```

Operation `UnSubscribe`

Die Operation `UnSubscribe` hat keine Ausgabe, da der Status *`1`* bedeutet, dass die Operation erfolgreich ausgeführt wurde.

Alive Listener Konfiguration

Es gibt einen Alive Listener, der es ermöglicht, Aktionen für jede Live-Benachrichtigung eines Abonnements zu konfigurieren.

Dabei lassen sich Prozesse für folgende Situationen definieren:

- `After create`: Das Abonnement wurde auf dem Server (neu) erstellt. Dies ist der Fall, wenn die Abonnierung aufgrund einer Kommunikationsunterbrechung in einen Timeout geriet und nach der Wiederverbindung neu erstellt werden muss.
- `Alive`: Der Server bestätigte, dass die Verbindung aktiv ist, obwohl es keine Änderungen zu senden gab.
- `Life timeout`: Der Server hat nicht bestätigt, dass die Verbindung aktiv ist, und der Life-Timeout wurde überschritten.
- `Timeout`: Der Server hat nicht bestätigt, dass die Verbindung aktiv ist, und die maximale Keepalive-Anzahl wurde überschritten.

Die Aktionen werden als Teil des Input-XML-Dokuments wie folgt definiert:

Input for Subscribe operation

```
<?xml version="1.0" encoding="UTF-8"?>
<Subscriptions>
    <Subscription ...></Subscription>
    <Subscription ...></Subscription>

    <!-- Alive listener actions  -->
    <!-- We can define some of them or none, they are not required -->

    <OnAfterCreate>x4db://1/TestingOPCUA/Processes/OnAfterCreate.wrf</
OnAfterCreate>
    <OnAlive>x4db://1/TestingOPCUA/Processes/OnAlive.wrf</OnAlive>
    <onLifetimeTimeout>x4db://1/TestingOPCUA/Processes/
onLifetimeTimeout.wrf</onLifetimeTimeout>
    <onTimeout>x4db://1/TestingOPCUA/Processes/onTimeout.wrf</onTimeout>
</Subscriptions>
```

10.13.6 SIMATIC S7 Connector

`SIMATIC S7 Connector` : Ermöglicht das Anbinden von *Siemens SIMATIC S7*-Systemen als SPS aus Technical Processes heraus. Dabei lassen sich sowohl Daten aus der Steuerung auslesen als auch an diese übermitteln. Über eine XML-Input-Struktur kann bestimmt werden, welche Daten gelesen bzw. geschrieben werden sollen. Es können *SIMATIC*-Systeme der Reihe S7-200, 300, 400 angebunden werden.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • `Read`: Daten aus der SPS auslesen • `Write`: Daten in die SPS schreiben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.s7.S7Connector:* Hauptklasse (Standard)
host	Adresse, über welche die SPS im Netzwerk erreichbar ist **Mögliche Werte:** IP-Adresse oder Host-Name

rack	Rack, das auf der SPS angesprochen werden soll **Mögliche Werte:** Nummer des Racks (beim SPS-Entwickler zu erfragen)
slot	Slot, der auf der SPS angesprochen werden soll **Mögliche Werte:** Nummer des Slots (beim SPS-Entwickler zu erfragen)

Statuswerte

1 (ok)	Erfolgreiche Kommunikation (Lesen oder Schreiben) mit der SPS
-1 (error)	Fehler bei der Kommunikation (Lesen oder Schreiben) mit der SPS Eine Fehlermeldung wird zurückgegeben

Input

Dieser Adapter erwartet jeweils ein XML-Dokument mit folgender Input-Struktur:

Input: Operation Read

```
<Read dbnum="[DBNUM]" bytes="[LENGTH]" offset="[OFFSET]">
    <FELDNAME s7type="[TYP]" s7size="[LENGTH]" />
    ...
</Read>
```

Parameter:

- `dbnum`: DBNummer von der gelesen werden soll (beim SPS-Entwickler zu erfragen)
- `bytes`: Anzahl der zu lesenden Bytes
- `offset`: Offset in Bytes, ab dem gelesen werden soll
- Die Elemente `FELDNAME` sind frei wählbar
- `s7type`: Typ des zu lesenden Feldes;
  Mögliche Werte:
  - `Array`
  - `Bit` **(Hinweis:** liest nur 1 Bit, daher müssen immer 8 Felder angegeben werden!)
  - `Byte`
  - `DateAndTime`
  - `Date` (im Format `yyyy-MM-dd'T'HH:mm:ss`)
  - `Integer`
  - `Long`
  - `Real`
  - `String`
  - `Time` (als Long)

- `s7size`: Länge in Bytes, die für das Feld gelesen werden sollen. Nur für die Typen `Array` und `String` notwendig

Input: Operation Write

```
<Write dbnum="DBNUM" bytes="LENGTH" offset="OFFSET">
    <FELDNAME s7type="String" s7size="12">[WERT]</AUFNR>
    ...
</Write>
```

Parameter:

- siehe Operation `Read`
- Der `WERT` des Elements muss syntaktisch zum Typ passen

Ausgabe

Dieser Adapter gibt ein XML-Dokument mit folgendem Aufbau zurück:

- Operation `Read`: Die Ausgabe entspricht der Eingabe, die Elemente sind bei Erfolg jeweils mit den passenden Werten gefüllt
- Operation `Write`: Die Ausgabe entspricht der Eingabe

10.14 Message Oriented Middleware

- Apache Kafka
- AMQP Publisher/Subscriber
- JMS
- Websphere MQ

10.14.1 Apache Kafka

Apache Kafka ermöglicht es, Datenströme über eine verteilte Streaming-Plattform zu speichern und zu verarbeiten. Mit verschiedenen Schnittstellen können Daten in Kafka-Cluster geschrieben, Daten gelesen oder aus Drittsystemen im- oder exportiert werden. Durch die hohe Fehlertoleranz und Skalierbarkeit eignet sich Apache Kafka für große Datenmengen und Anwendungen z. B. im Big-Data-Umfeld.

> Weitere Informationen zur Verwendung von Apache Kafka in einer lokalen Umgebung finden sich hier.
> Weitere Informationen zur Konfiguration und Integration von TLS-Zertifikaten in Apache Kafka Cluster finden sich hier.

Um die beiden Adapter Apache Kafka Publisher bzw. Apache Kafka Subscriber zu verwenden, wird eine Konfigurationsdatei in der X4 Server-Installation benötigt. Diese Konfigurationsdatei muss `apachekafka-config.xml` heißen und im Ordner `X4DB/0` liegen. Beim Start des X4 Servers werden alle Informationen aus dieser Datei geladen.

Die folgende XML-Datei zeigt alle erforderlichen Elemente und Attribute in der richtigen Struktur:

Beispieldatei apachekafka-config.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<ApacheKafkaConfiguration>
    <!-- Many clusters can be defined here -->
    <Cluster id="ApacheKafka_broker_1" server="localhost:9092">

        <!-- Now the different clients that can work with this broker are defined:
                For each ClientId, we define the following things:
                - The ClientId name: alias to identify the client in X4 Processes.
                - groupId: the id of the group where the consumer/publisher belongs
to.
        -->
        <Client id = "AK_Client_1" groupId="test" autoCommitIntervalMS="100"/>
        <Client id = "AK_Client_2" groupId="test" autoCommitIntervalMS="100"/>

        <!-- Now, the different topics that clients can interact with for this broker
are defined  -->
        <Topic filter="topic1"/>
        <Topic filter="topic2"/>
        <Topic filter="topic3"/>
    </Cluster>

    <Cluster id="ApacheKafka_broker_2" server="localhostel:9093">

        <!-- Node that includes the SSL/TLS needed configuration for server  -->
        <SslConfiguration privateKeyPassword="password">
            <!-- Server keystore password and server keystore path -->
            <KeyStore password="password" path="C:\tmp\ssl_new\server.keystore.jks"/>
            <!-- Server truststore password and server truststore path -->
            <TrustStore password="password" path="C:
\tmp\ssl_new\server.truststore.jks"/>
        </SslConfiguration>

        <!-- Now the different clients that can work with this broker are defined:
                For each ClientId, we define the following things:
                - The ClientId name: alias to identify the client in X4 Processes.
                - groupId: the id of the group where the consumer/publisher belongs
to.
        -->
        <Client groupId="test" autoCommitIntervalMS="100" id = "AK_Client_3"/>
        <Client groupId="test" autoCommitIntervalMS="100" id = "AK_Client_4"/>

        <!-- Now, the different topics that clients can interact with for this broker
are defined  -->
        <Topic filter="topic2"/>
        <Topic filter="topic5"/>
        <Topic filter="topic4"/>
    </Cluster>
</ApacheKafkaConfiguration>
```

Im Folgenden sind die wichtigsten Elemente und Attribute näher erläutert:

Element / Attribut	Beschreibung
`Cluster`	*Muss eindeutig sein*. Definiert die erforderlichen Parameter, um die Verbindung zu einem bestimmten Kafka-Cluster zu konfigurieren. Im Attribut `id` befindet sich die Information, die für den Adapter-Parameter `clusterId` benötigt wird.
`server`	URL des Clusters. Unterstützt das TCP-Protokoll. **Mögliche Werte.** `hostname:port`
`SslConfiguration`	Definiert die SSL/TLS-Konfigurationsparameter.
`privateKeyPassword`	Passwort des Private Key in der Keystore-Datei.
`KeyStore`	Definiert, wo sich der zu verwendende Keystore befindet und welches Passwort hinterlegt ist.
`TrustStore`	Definiert, wo sich der zu verwendende Trust Store befindet und welches Passwort hinterlegt ist.
`ClientId`	Informationen zum Client, der verwendet wird, um Nachrichten zu einem bestimmten Topic zu veröffentlichen.
`groupId`	Definiert die Gruppen-ID, die verwendet wird, um die Verbindungen zu gruppieren.
`autoCommitIntervalMS`	Definiert die Häufigkeit der Commits in Millisekunden. ⓘ Dieses Attribut wird für den Adapter Apache Kafka Subscriber benötigt.
`Topic`	Topic, in dem die Nachrichten veröffentlicht werden.
`filter`	Name des Topics

ⓘ Die Werte für `KeyStore`, `TrustStore` und `privateKeyPassword` können auch über Adapterparameter gesetzt werden. Werte, die über Adapterparameter gesetzt werden, überschreiben die Werte aus der `apachekafka-config.xml`.

10.14.1.1 Apache Kafka Publisher

`Apache Kafka Publisher`: Ermöglicht es, Nachrichten zu einem bestimmten Apache Kafka Topic zu veröffentlichen.
Eigenschaften (Properties)

`Operation`	Definiert die vom Adapter ausgeführte Operation. **Mögliche Werte:** • `Publish`: Veröffentlicht eine Nachricht zu einem bestimmten Topic, die in einem XML-Input angegeben ist.

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.apachekafka.ApacheKafkaPublisherAdapter*: Hauptklasse (Standard)
`trustStorePath`	Definiert, wo sich der zu verwendende Trust Store befindet.
`trustStorePassword`	Definiert, welches Passwort für den Trust Store verwendet wird.
`keyStorePath`	Definiert, wo sich der zu verwendende Key Store befindet.
`keyStorePassword`	Passwort für den Key Store
`privateKeyPassword`	Passwort des Private Key in der Key-Store-Datei
`username`	Benutzername zur SASL/PLAIN-Authentifizierung am Kafka-Server ⚠ Wenn Sie diesen Parameter verwenden, werden die Konfigurationseigenschaften `sasl.mechanism = PLAIN` und `security.protocol = SASL_SSL` gesetzt.
`password`	Password zur SASL/PLAIN-Authentifizierung am Kafka-Server ⚠ Wenn Sie diesen Parameter verwenden, werden die Konfigurationseigenschaften `sasl.mechanism = PLAIN` und `security.protocol = SASL_SSL` gesetzt.
`clusterId`	ID des Clusters, in dem die Nachricht veröffentlicht wird **Mögliche Werte:** Cluster-ID ⓘ Die ID des Clusters kann in der Datei `apachekafka-config.xml` gefunden werden. Sie befindet sich im Attribut `id` des Elements `Cluster`.

`clientId`	ID des Clients, mit dem die Nachricht veröffentlicht wird **Mögliche Werte:** Client-ID ⓘ Die ID des Clients kann in der Datei `apachekafka-config.xml` gefunden werden. Sie befindet sich im Attribut `id` des Elements `Client`.
`topic`	Name des Topics, in dem die Nachricht veröffentlicht wird **Mögliche Werte:** Zeichenkette aus alphanumerischen ASCII-Zeichen, `.` (Punkt), _ (Unterstrich) und – (Bindestrich)
`messageDataType`	Datentyp der Nachricht(en), die veröffentlicht werden. **Mögliche Werte:** • `BYTES`: Base64-kodierter Wert • `BYTEARRAY`: Base64-kodierter Wert • `DOUBLE`: Gleitkommazahl • `FLOAT`: Gleitkommazahl • `INTEGER`: Ganzzahl • `LONG`: Ganzzahl • `SHORT`: Ganzzahl • `STRING` (Standard): Zeichenkette • `UUID`: Java UUID ⚠ In einem Topic können Nachrichten mit verschiedenen Datentypen veröffentlicht werden. Um die Nachrichten mit dem Apache Kafka Subscriber Adapter zu lesen, muss der Datentyp bekannt und korrekt eingestellt sein. Beispielsweise kann eine Nachricht vom Datentyp `STRING` nicht mit der Datentyp-Einstellung `LONG` gelesen werden. Daher sollten alle Nachrichten in einem Topic vom selben Datentyp sein.

`keyDataType`	Datentyp des Schlüssels
	
	Mögliche Werte:
	
	• `NONE` (Standard): Dieser Wert wird verwendet, wenn kein Schlüssel definiert ist.
	• `BYTES`: Base64-kodierter Wert
	• `BYTEARRAY`: Base64-kodierter Wert
	• `DOUBLE`: Gleitkommazahl
	• `FLOAT`: Gleitkommazahl
	• `INTEGER`: Ganzzahl
	• `LONG`: Ganzzahl
	• `SHORT`: Ganzzahl
	• `STRING` (Standard): Zeichenkette
	• `UUID`: Java UUID
	
	Beispielhafte Schlüssel-Definition
	
	`<Messages>`
	`<Message key="055b93a8-4eda-41d8-8b73-b852a1d4da1e">055b93a8-4eda-41d8-8b73-b852a1d4da1e</Message>`
	`</Messages>`
	
	Folgende vier Fälle können unterschieden werden:
	
	• Ein Schlüssel mit einem anderen Datentyp als `NONE` wurde definiert: Die Nachricht wird mit einem Schlüssel im angegebenen Datentyp versendet.
	• Ein Schlüssel mit dem Datentyp `NONE` wurde definiert: Die Nachricht wird ohne Schlüssel versendet und eine Warnung wird im Server-Log ausgegeben.
	• Es wurde kein Schlüssel definiert und der Datentyp des Schlüssels hat einen anderen Wert als `NONE`: Die Nachricht wird nicht versendet und eine Fehlermeldung wird ausgegeben.
	• Es wurde kein Schlüssel definiert und der Datentyp des Schlüssels ist `NONE`: Die Nachricht wird ohne Schlüssel versendet.

Die Werte der Parameter `trustStorePath`, `trustStorePassword`, `keyStorePath`, `keyStorePassword` und `privateKeyPassword` können auch in der Datei `apachekafka-config.xml` definiert werden (siehe Apache Kafka).

⚠ Die Angaben über die Adapterkonfiguration überschreiben die Angaben in der `apachekafka-config.xml`. Dafür müssen aber **alle** Parameter gesetzt werden. Wenn nicht alle Parameter über die Adapterkonfiguration gesetzt werden, werden die Dateien aus der `apachekafka-config.xml` verwendet.

⚠ Wenn Adapterparameter verwendet werden, um die SSL-Konfiguration eines bestimmten Clients zu überschreiben, dann werden diese Werte beibehalten, bis sie erneut überschrieben oder der Server neugestartet wird.

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters kam es zu einem Fehler.

Input

Der Adapter erwartet eine vordefinierte XML-Struktur als Input. Die Struktur ist abhängig vom Datentyp der Nachricht.

- Datentyp `STRING`:

```
<Messages>
    <Message>
        <!-- This is a message. -->
    </Message>
    <Message>
        <!-- This is another message. -->
    </Message>
</Messages>
```

- Datentyp `BYTES`:

```
<Messages>
    <Message>ZXN0byBlcyB1biBmaWNoZXJvIGRlIHRleHRv</Message>
</Messages>
```

- Datentyp `NUMERIC`:

```
<Messages>
    <Message>5428</Message>
</Messages>
```

Nachrichten mit und ohne Header können beliebig vermischt werden:

```
<?xml version="1.0" encoding="UTF-8"?>
<Messages>
<!-- Example 1: A message with many headers -->
    <Message key="header-message">
        <Headers>
            <Header key="header1">value 1</Header>
            <Header key="header2">value 2</Header>
            <Header key="header3">value 3</Header>
        </Headers>
        <Body>Successfull message</Body>
    </Message>
<!-- Example 2: A message without headers but with message key -->
    <Message key="header-message">
        <Body>Successfull message</Body>
    </Message>
<!-- Example 3: A message without headers and message key -->
    <Message>
        <Body>Successfull message</Body>
    </Message>
</Messages>
```

Output

Bei einer erfolgreichen Ausführung wird der Input als Output ausgegeben.

10.14.1.2 Apache Kafka Subscriber

`Apache Kafka Subscriber`: Ermöglicht es, Nachrichten zu einem bestimmten Topic zu abonnieren, das Abonnement zu beenden oder die letzte Nachricht zu lesen.
Eigenschaften (Properties)

`Operation`	Definiert die vom Adapter ausgeführte Operation. **Mögliche Werte:** • `Subscribe`: Abonniert ein Topic, um Nachrichten zu erhalten. Dabei wird ein vorhandener Client aus der Datei `apachekafka-config.xml` verwendet. ⚠ Für ein Abonnement kann nur ein Client verwendet werden, der noch nicht für ein anderes Topic verwendet wird. Ein Client kann nicht mehrere Topics abonnieren. **Erforderliche Parameter:** `clusterId`, `clientId`, `topic`, `process` • `Unsubscribe`: Trennt Client und Topic und beendet das Abonnement eines bestimmten Topics. **Erforderliche Parameter:** `clusterId`, `clientId` • `Last offset`: Gibt den Offset eines bestimmten Clients bei dessen abonnierten Topic zurück. Mit dem Offset können vergangene Nachrichten erneut empfangen werden. ⚠ Wenn diese Operation für einen Client verwendet wird, der kein Topic abonniert hat, dann wird ein leeres Ergebnis zurückgegeben. **Erforderliche Parameter:** `clusterId`, `clientId`, `topic`

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.apachekafka.ApacheKafkaSubscriberAdapter*: Hauptklasse (Standard)
`trustStorePath`	Definiert, wo sich der zu verwendende Trust Store befindet.
`trustStorePassword`	Definiert, welches Passwort für den Trust Store verwendet wird.
`keyStorePath`	Definiert, wo sich der zu verwendende Key Store befindet.
`keyStorePassword`	Passwort für den Key Store
`privateKeyPassword`	Passwort des Private Key in der Key-Store-Datei
`username`	Benutzername zur SASL/PLAIN-Authentifizierung am Kafka-Server ⚠ Wenn Sie diesen Parameter verwenden, werden die Konfigurationseigenschaften `sasl.mechanism = PLAIN` und `security.protocol = SASL_SSL` gesetzt.

`password`	Password zur SASL/PLAIN-Authentifizierung am Kafka-Server ⚠ Wenn Sie diesen Parameter verwenden, werden die Konfigurationseigenschaften `sasl.mechanism = PLAIN` und `security.protocol = SASL_SSL` gesetzt.
`clusterId`	ID des Clusters, mit dem das Topic abonniert wird **Mögliche Werte:** Cluster-ID ⓘ Die ID des Clusters kann in der Datei `apachekafka-config.xml`gefunden werden. Sie befindet sich im Attribut `id` des Elements `Cluster`.
`clientId`	ID des Clients, mit dem das Topic abonniert wird **Mögliche Werte:** Client-ID ⓘ Die ID des Clients kann in der Datei `apachekafka-config.xml` gefunden werden. Sie befindet sich im Attribut `id` des Elements `Client`.
`topic`	Name des Topics, das abonniert wird **Mögliche Werte:** Zeichenkette aus alphanumerischen ASCII-Zeichen, `.` (Punkt), _ (Unterstrich) und – (Bindestrich) oder kommagetrennte Liste aus mehreren Zeichenketten, um mehrere Topics zu abonnieren
`process`	Prozess, der ausgeführt wird, nachdem eine Nachricht empfangen wurde. **Mögliche Werte:** Zeichenkette aus alphanumerischen ASCII-Zeichen, `.` (Punkt), _ (Unterstrich) und – (Bindestrich)
`consumerPosition`	Offset, mit dem Nachrichten empfangen werden, wenn ein Topic abonniert wird. **Mögliche Werte:** Ganzzahl größer als `0` ⚠ Dieser Parameter hat nur einen Effekt, wenn ein einzelnes Topic abonniert wird. Wenn mehrere Topics abonniert werden, hat dieser Parameter keine Auswirkungen.

`messageDataType`	Datentyp der Nachricht(en), die veröffentlicht werden. **Mögliche Werte:** • `BYTES`: Base64-kodierter Wert • `BYTEARRAY`: Base64-kodierter Wert • `DOUBLE`: Gleitkommazahl • `FLOAT`: Gleitkommazahl • `INTEGER`: Ganzzahl • `LONG`: Ganzzahl • `SHORT`: Ganzzahl • `STRING` (Standard): Zeichenkette • `UUID`: Java UUID ⚠ In einem Topic können Nachrichten mit verschiedenen Datentypen veröffentlicht werden. Um die Nachrichten mit dem Apache Kafka Subscriber Adapter zu lesen, muss der Datentyp bekannt und korrekt eingestellt sein. Beispielsweise kann eine Nachricht vom Datentyp `STRING` nicht mit der Datentyp-Einstellung `LONG` gelesen werden. Daher sollten alle Nachrichten in einem Topic vom selben Datentyp sein.
`keyDataType`	Datentyp des Schlüssels **Mögliche Werte:** • `NONE` (Standard): Kein Datentyp ⓘ Hat der Subscriber ein Topic abonniert und der `keyDataType` hat den Wert `NONE`, werden ausschließlich Nachrichten ohne Schlüssel empfangen. Ist der Wert ein anderer als `NONE`, werden Nachrichten mit allen anderen Datentypen empfangen. Hat die Nachricht einen Schlüssel, wird der Schlüssel als Attribut im Element `Content` ausgegeben. • `BYTES`: Base64-kodierter Wert • `BYTEARRAY`: Base64-kodierter Wert • `DOUBLE`: Gleitkommazahl • `FLOAT`: Gleitkommazahl • `INTEGER`: Ganzzahl • `LONG`: Ganzzahl • `SHORT`: Ganzzahl • `STRING` (Standard): Zeichenkette • `UUID`: Java UUID

Die Werte der Parameter `trustStorePath`, `trustStorePassword`, `keyStorePath`, `keyStorePassword` und `privateKeyPassword` können auch in der Datei `apachekafka-config.xml` definiert werden (siehe Apache Kafka).

Die Angaben über die Adapterkonfiguration überschreiben die Angaben in der `apachekafka-config.xml`. Dafür müssen aber **alle** Parameter gesetzt werden. Wenn nicht alle Parameter über die Adapterkonfiguration gesetzt werden, werden die Dateien aus der `apachekafka-config.xml` verwendet.

Wenn Adapterparameter verwendet werden, um die SSL-Konfiguration eines bestimmten Clients zu überschreiben, dann werden diese Werte beibehalten, bis sie erneut überschrieben oder der Server neugestartet wird.

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters kam es zu einem Fehler.

Input

Dieser Adapter erwartet keinen Input.

Output

Der Adapter gibt abhängig von der ausgeführten Operation folgenden Output aus:

- Operation `Subscribe`: Für diese Operation wird kein direkter Output ausgegeben. Wenn in einem abonnierten Topic eine neue Nachricht empfangen wird, dann wird ein Prozess mit der empfangenen Nachricht als Input gestartet. Diese Nachricht wird in folgender Struktur übergeben:

Beispiel Output-Operation Subscribe

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Message>
    <ClusterId>X4Kafka</ClusterId>
    <Topic>noKeyTopic</Topic>
    <Offset>0</Offset>
    <Headers>
        <Header key="value 1">header1</Header>
        <Header key="value 2">header2</Header>
        <Header key="value 3">header3</Header>
    </Headers>
    <Content encoding="base64"key="055b93a8-4eda-41d8-8b73-
b852a1d4da1e">055b93a8-4eda-41d8-8b73-b852a1d4da1e</Content>
</Message>
```

- Operation `Unsubscribe`: Kein Output
- Operation `Last offset`:

Beispiel Output Operation Last offset

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Offsets>
    <Offset partitionId="0" value="17"/>
</Offsets>
```

Im Output dieser Adapteroperation zeigt der Wert von `partitionId` die Partition, in der sich das Topic befindet, und der Wert von `value` den Offset zur letzten Nachricht, die der angegebene Client gelesen hat.

10.14.2 AMQP Publisher/Subscriber

10.14.2.1 AMQP Publisher

`AMQP Publisher`: Ermöglicht das Veröffentlichen von Nachrichten in einem AMQP-Topic.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Publish`: Nachrichten in einem AMQP-Topic veröffentlichen

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.amqp.AmqpPublisherAdapter* : Hauptklasse (Standard)
`storedConfig`	Bezeichner der Konfiguration, bereitgestellt in der Datei `amqpconfig.xml` **Mögliche Werte:** Beliebige Zeichenkette mit dem Bezeichner, z. B. *`queueEmployeesConfig`*
`host`	AMQP Server-Host **Mögliche Werte:** Server-Host, z. B. *`localhost`*
`port`	Dedizierter AMQP-Port auf dem Server **Mögliche Werte:** AMQP-Port-Nummer, z. B. *`5672`*
`username`	Name des Benutzers, der die Nachricht veröffentlicht **Mögliche Werte:** Beliebige Zeichenkette mit dem Benutzernamen, z. B. *`user`*
`password`	Password des Benutzers **Mögliche Werte:** Beliebige Zeichenkette mit dem Passwort, z. B. *`p4ssw0rd`*
`queue`	Warteschlange, in der die Nachricht veröffentlicht wird **Mögliche Werte:** Beliebige Zeichenkette mit dem Namen der Warteschlange, z. B. *`employees`*
`container`	Container, in dem sich die Warteschlange befindet **Mögliche Werte:** Beliebige Zeichenkette mit dem Container-Namen z. B. *`employeesContainer`*

`qos`	Quality of Service der Nachricht **Mögliche Werte:** • `AT_MOST_ONCE`: Die Nachricht wird höchstens ein Mal oder kein einziges Mal zugestellt. • `AT_LEAST_ONCE`: Die Nachricht wird mindestens ein Mal zugestellt. • `EXACTLY_ONCE`: Die Nachricht wird nur ein Mal zugestellt. Ist der Parameter `qos` nicht angegeben, ist der Standard-Wert `AT_MOST_ONCE`.
`useSsl`	SSL-Verschlüsselung aktivieren **Mögliche Werte:** `true` / `false` (Standard)
`useSasl`	SASL-Verschlüsselung aktivieren **Mögliche Werte:** • `true` • `false` (Standard)
`mechanism`	SASL-Mechanismus des entfernten Endpoints **Mögliche Werte:** • `PLAIN`: Alle Daten werden im Klartext ausgetauscht (Standard). • `ANONYMOUS`: Der Dienst kann ohne Authentifizierung genutzt werden. • `CRAM-MDS`: Verhindert die Passwort-Übertragung im Klartext. • `Digest-MDS`: Zusätzliche Parameter wie Integritätssicherung können verwendet werden.
`durable`	Nachricht als dauerhaft definieren **Mögliche Werte:** `true` (Standard) / `false`
`priority`	Prioritätsstufe der Nachricht **Mögliche Werte:** Beliebige Ganzzahl zwischen `0` und `255`, z. B. *`3`*
`ttl`	Zeitspanne in Millisekunden, nach der die Nachricht verworfen werden soll **Mögliche Werte:** Beliebige Ganzzahl, die die Zeitspanne in Millisekunden angibt, z. B. *`4500`*

Statuswerte

`0`	Die Operation wurde erfolgreich ausgeführt (es wird keine Ausgabe generiert)

`-1`	Während der Ausführung der Operation kam es zu einem Fehler (es wird keine Ausgabe generiert)

Konfiguration

Um die benötigten Informationen bereitzustellen, kann eine Konfigurationsdatei verwendet werden, die beim Serverstart geladen wird. Die Konfigurationsdatei `amqpconfig.xml` muss unter dem Pfad `X4DB/0` bereitgestellt werden.

Die Datei hat folgende Struktur:

amqpconfig.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<RootElement>
    <AMQPConfig name="queueEmployeesConfig">
        <Server host="localhost" port="5672" useSsl="false"/>
        <Client user="user" password="p4ssw0rd"/>
        <Queue container="employeesContainer" name="employees" qos="AT_LEAST_ONCE"/>
    </AMQPConfig>
</RootElement>
```

Input

Der AMQP Publisher erwartet folgende Input-Struktur, wobei nur ein Element pro Nachricht erlaubt ist:

Einfache Typen:

Beispiel-Input: Einfache Typen

```
<Messages>
    <Message>
        <Element type="String" value="Good morning!!"/>
    </Message>
    <Message>
        <Element type="Int" value="18"/>
    </Message>
</Messages>
```

Folgende einfache Typen sind erlaubt: `Binary`, `Boolean`, `Byte`, `Char`, `Decimal32`, `Decimal64`, `Decimal128`, `Double`, `Float`, `Long`, `Short`, `String`, `Symbol`, `Timestamp` (nach dem Muster *`yyyy-[m]m-[d]d hh:mm:ss`*), `UByte`, `UInt`, `ULong`, `UShort`, `Uuid`.

Komplexe Typen:

- Liste:

Beispiel-Input: Liste

```
<Messages>
    <Message>
        <Element type="List">
            <Element type="Int" value="1"/>
            <Element type="Short" value="3"/>
            <Element type="Boolean" value="false"/>
        </Element>
    </Message>
</Messages>
```

- Map:

Beispiel-Input: Map

```
<Messages>
    <Message>
        <Element type="Map">
            <Entries>
                <Entry>
                    <Key type="Int" value="2"/>
                    <Value type="String" value="Hello!!"/>
                </Entry>
                <Entry>
                    <Key type="Int" value="5"/>
                    <Value type="String" value="Goodbye!!"/>
                </Entry>
            </Entries>
        </Element>
    </Message>
</Messages>
```

10.14.2.2 AMQP Subscriber

`AMQP Subscriber`: Verwaltet Abonnements für eine AMQP-Warteschlange und definiert den auszuführenden Prozesses.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Subscribe`: AMQP-Warteschlange abonnieren und den auszuführenden Prozess definieren Erforderliche Parameter sind `host`, `port`, `username`, `password`, `queue` und `process`. Die Parameter können auch über eine Konfigurationsdatei bereitgestellt werden, die im Parameter `storedConfig` referenziert wird. • `Unsubscribe`: AMQP-Warteschlange abkündigen Erforderliche Parameter sind `username` und `queue`. Die Parameter können auch über eine Konfigurationsdatei bereitgestellt werden, die im Parameter `storedConfig` referenziert wird.

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.amqp.AmqpSubscriberAdapter*: Hauptklasse (Standard)
`storedConfig`	Bezeichner der Konfiguration, bereitgestellt in der Datei `amqpconfig.xml` **Mögliche Werte:** Beliebige Zeichenkette mit dem Bezeichner, z. B. *`queueEmployeesConfig`*
`host`	AMQP Server-Host **Mögliche Werte:** Server-Host, z. B. *`localhost`*
`port`	Dedizierter AMQP-Port auf dem Server **Mögliche Werte:** AMQP-Port-Nummer, z. B. *`5672`*
`username`	Name des Benutzers, der die Nachricht abonniert bzw. abkündigt **Mögliche Werte:** Beliebige Zeichenkette mit dem Benutzernamen, z. B. *`user`*
`password`	Password des Benutzers **Mögliche Werte:** Beliebige Zeichenkette mit dem Passwort, z. B. *`p4ssw0rd`*

`queue`	Warteschlange, in der die Nachricht abonniert bzw. abkündigt wird **Mögliche Werte:** Beliebige Zeichenkette mit dem Namen der Warteschlange, z. B. *`employees`*
`container`	Container, in dem sich die Warteschlange befindet **Mögliche Werte:** Beliebige Zeichenkette mit dem Container-Namen z. B. *`employeesContainer`*
`qos`	Quality of Service der Nachricht **Mögliche Werte:** • `AT_MOST_ONCE`: Die Nachricht wird höchstens ein Mal oder kein einziges Mal zugestellt. • `AT_LEAST_ONCE`: Die Nachricht wird mindestens ein Mal zugestellt. • `EXACTLY_ONCE`: Die Nachricht wird nur ein Mal zugestellt. Ist der Parameter `qos` nicht angegeben, ist der Standard-Wert `AT_MOST_ONCE`.
`useSsl`	SSL-Verschlüsselung aktivieren **Mögliche Werte:** `true` / `false` (Standard)
`useSasl`	SASL-Verschlüsselung aktivieren **Mögliche Werte:** • `true` • `false` (Standard)
`mechanism`	SASL-Mechanismus des entfernten Endpoints **Mögliche Werte:** • `PLAIN`: Alle Daten werden im Klartext ausgetauscht (Standard). • `ANONYMOUS`: Der Dienst kann ohne Authentifizierung genutzt werden. • `CRAM-MDS`: Verhindert die Passwort-Übertragung im Klartext. • `Digest-MDS`: Zusätzliche Parameter wie Integritätssicherung können verwendet werden.
`process`	Pfad des Technical Processes, der ausgeführt wird, sobald eine Nachricht empfangen wird **Mögliche Werte:** Prozesspfad, z. B. *`x4db://1/Project/Processes/process.wrf`*

Statuswerte

`0`	Die Operation wurde erfolgreich ausgeführt (es wird keine Ausgabe generiert)
`-1`	Während der Ausführung der Operation kam es zu einem Fehler (es wird keine Ausgabe generiert)

Konfiguration

Um die benötigten Informationen bereitzustellen, kann eine Konfigurationsdatei verwendet werden, die beim Serverstart geladen wird. Die Konfigurationsdatei `amqpconfig.xml` muss unter dem Pfad `X4DB/0` bereitgestellt werden.

Die Datei hat folgende Struktur:

amqpconfig.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<RootElement>
    <AMQPConfig name="queueEmployeesConfig">
        <Server host="localhost" port="5672" useSsl="false"/>
        <Client user="user" password="p4ssw0rd"/>
        <Queue container="employeesContainer" name="employees" qos="AT_LEAST_ONCE"/>
    </AMQPConfig>
</RootElement>
```

10.14.3 JMS

`JMS`: Tauscht Nachrichten über die *Java Message Service*-Warteschlange eines nachrichtenorientierten Middleware-Produkts aus, das als JMS-Provider dient. Der Adapter unterstützt das Senden und Empfangen von Nachrichten von und zur JMS-Warteschlange und kann Nachrichten in JMS-Topics publizieren und JMS-Topics abonnieren.

Ehemalige Bezeichnung bis Version 4.5.1: *JMS Transfer Adapter*
Ehemalige Bezeichnung bis Version 5.5.4: *JMS Connector*

Eigenschaften (Properties)

Delete	(keine Funktion)
File name	(keine Funktion)
I/O Mode	(keine Funktion)

Operation	Bestimmt, welche Operation mit den Nachrichten der JMS-Warteschlange durchgeführt wird. **Mögliche Werte:** • `Send`: Sendet Daten aus dem vorangegangen Prozess-Schritt an die JMS-Warteschlange • `Receive`: Liest JMS-Nachricht in den Prozess ein • `List`: Gibt die Zahl der Nachrichten in der Warteschlange als XML-Dokument aus (falls Parameter `toXML` gesetzt ist, werden stattdessen die Header-Informationen der Nachrichten ausgegeben).

Parameter

object_name	Name der JMS-Warteschlange oder des JMS-Topics, unter dem es im JNDI-Provider (Java Naming and Directory Interface) registriert ist **Mögliche Werte:** Beliebige Zeichenkette (z. B. *`queue/A`*)
connection_factory	Name der *JMS Connection Factory* , unter der die Nachrichtenwarteschlange im JNDI (Java Naming and Directory Interface) registriert ist **Mögliche Werte:** Beliebige gültige Zeichenkette (z. B. *`java:/JmsXA`*)
JNDIProperties	Zusätzliche Eigenschaften des JNDI-Providers **Mögliche Werte:** Beliebige Zeichenkette mit Eigenschaften, bei mehreren Eigenschaften durch Semikolon getrennt (z. B. *java.naming.factory.initial=org.jnp.inter faces.Naming ContextFactory;java.naming.provi der.url=localhost:1099*)
approach	Verwendeter Nachrichtenkanaltyp **Mögliche Werte:** • `topic`: Punkt-zu-Punkt-Verbindung zwischen einem Sender und einem Empfänger, welcher nach Erhalt die Nachricht normalerweise aus dem Nachrichtenkanal entfernt • `queue`: Themenorientierte Publizieren/Abonnieren-Verbindung, bei der Nachrichten im Nachrichtenkanal erhalten bleiben, um auch von anderen Empfängern abgerufen werden zu können, bis die Nachrichten ersetzt werden oder verfallen (Standard)

timeout	Zeitbeschränkung für den Verbindungsaufbau in Millisekunden **Mögliche Werte:** • Beliebige ganze Zahl • `0`: keine Beschränkung; Vorsicht: Wenn die Warteschlange leer ist, dann "steht" der Prozess mit dieser Einstellung! • `10000`: Maximal 10 Sekunden warten (Standard)
filter	Filtert beim Empfangen von Nachrichten über JMS-Nachrichtenselektoren **Mögliche Werte:** Beliebige Zeichenkette mit Eigenschaften nach dem Muster `<Eigenschaft1>=<Wert1>;<Eigenschaft2>=<Wert2>` (z. B. *`JMSPriority=5`*)
msg_property	Header-Attribute der JMS-Nachricht beim Senden von Nachrichten **Mögliche Werte:** Beliebige Zeichenkette mit Eigenschaften nach dem Muster `<Eigenschaft1>=<Wert1>;<Eigenschaft2>=<Wert2>`
headers_from_XML	Header-Informationen des Input-Dokuments verwenden **Mögliche Werte:** • `ja`: Header-Informationen verwenden • `nein`: Keine Header-Informationen verwenden (Standard)
transacted	Diesen Prozess-Schritt in einer lokalen Transaktion ausführen **Mögliche Werte:** • `ja`: Als lokale Transaktion ausführen • `nein`: Keine lokale Transaktion ausführen (Standard)

ack_mode	Acknowledgement-Modus (dieser regelt den Umgang mit Empfangsbestätigungen) **Mögliche Werte:** • `AUTO_ACKNOWLEDGE|Auto acknowledge`: Automatisch den Empfang von Nachrichten bestätigen • `CLIENT_ACKNOWLEDGE|Client acknowledge`: Empfang durch einen Client bestätigen • `DUPS_OK_ACKNOWLEDGE|Dups ok acknowledge`: Mehrfaches Empfangen einer Nachricht erlauben
msg_priority	Nachrichten-Priorität (*9* hat die höchste Priorität) **Mögliche Werte:** • Beliebige ganze Zahlen von *0–9* • `4`: Priorität 4 (Standard)
msg_expiration_time	Gültigkeitsdauer von Nachrichten (Lebensdauer) in Millisekunden; nach dem Ablauf der Gültigkeit ist die Nachricht obsolet und wird automatisch aus der Warteschlange entfernt **Mögliche Werte:** • Beliebige positive ganze Zahl • `0`: keine Gültigkeitsbeschränkung (Standard)
msg_type	JMS-Nachrichtentyp **Mögliche Werte:** • `text`: Daten als Zeichenketten (etwa als XML-Dokumente) austauschen (Standard) • `byte`: Byte-Feldgruppe für Datenaustausch in nativen/ binären Formaten • `map`: JMS-MapMessages (Key-Value-Paare) versenden
subscription_name	Name des Abonnements für dauerhafte Nachrichten-Abonnements (wenn Parameter approach auf *topic* gesetzt wurde) **Mögliche Werte:** Beliebige Zeichenkette

durable_subscription	Dauer des JMS-Abonnements **Mögliche Werte:** • `ja`: dauerhaft beziehen • `nein`: temporär beziehen
ClientID	Name des JMS-Clients **Mögliche Werte:** Beliebige Zeichenkette
toXML	JMS-Nachricht inklusive Header-Informationen als XML-Dokument ausgeben (wenn Eigenschaft Operation auf *Receive* gesetzt wurde, dann wird der Zeichensatz unverändert übernommen; als MIME-Type wird dann *Text/XML* verwendet). Wenn Sie den Parameter toXML nicht setzen, dann werden Byte-,Stream- oder Object-Nachrichten als MIME-Typ *Application/Octet-stream* deklariert; Text-Nachrichten werden hingegen UTF-8-kodiert und mit MIME-Typ *Text/Plain* deklariert. **Mögliche Werte:** • `ja`: XML-Dokument ausgeben • `nein`: Kein XML-Dokument ausgeben (Standard)
base64	JMS-Nachricht Base64-kodiert ausgeben (wenn Parameter toXML gesetzt ist) **Mögliche Werte:** • `ja`: Nachricht Base64-kodiert ausgeben • `nein`: Keine Base64-kodierte Nachricht ausgeben (Standard)
JMSReplyTo	Ziel-Warteschlange der Antwort-Nachricht
JMSCorrelationID	ID der Nachricht, auf die sich die Antwort bezieht (verschiedene Teile einer Nachricht besitzen dieselbe JMSCorrelationID)
login	Benutzername für den JMS
password	Passwort für den JMS

Die im Applikations-Server standardmäßig verfügbaren JMS-Warteschlangen sind wegen des verwendeten Datenbanksystems (HSQLDB, Derby) nicht für den Produktiveinsatz geeignet. Trotz transaktionaler Verarbeitung können mit diesen Daten verloren gehen. Verwenden Sie ausschließlich produktiv einsetzbare Datenbanksysteme (Oracle, MSSQL etc.) zur Nachrichten-Persistierung von JMS-Warteschlangen; siehe http://docs.jboss.org/jbossas/docs/Server_Configuration_Guide/4/html/apbs04.html.

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input: msg:type `map`

```
<?xml version="1.0" encoding="UTF-8"?>
<RootElement>
    <Entry name="firstKey">Blub</Entry>
    <Entry name="b" type="byte">1</Entry>
    <Entry name="s" type="short">2</Entry>
    <Entry name="i" type="integer">3</Entry>
    <Entry name="l" type="long">4</Entry>
    <Entry name="f" type="float">5.5</Entry>
    <Entry name="d" type="double">6.6</Entry>
    <Entry name="bool" type="boolean">true</Entry>
    <Entry name="c" type="character">z</Entry>
    <Entry name="blob" type="bytearray">aaaa</Entry>
</RootElement>
```

Input: msg:type `map` und headers_from_XML

```
<?xml version="1.0" encoding="UTF-8"?>
<messages>
    <message type="map">
        <headers>
            <header>
                <name>myFirstHeader</name>
                <value>the first</value>
            </header>
            <header>
                <name>mySecondHeader</name>
                <value>the second</value>
            </header>
        </headers>
        <body>
            <Entry name="firstKey">Blub</Entry>
            <Entry type="byte" name="b">1</Entry>
            <Entry type="short" name="s">2</Entry>
            <Entry type="integer" name="i">3</Entry>
            <Entry type="long" name="l">4</Entry>
            <Entry type="float" name="f">5.5</Entry>
            <Entry type="double" name="d">6.6</Entry>
            <Entry type="boolean" name="bool">true</Entry>
            <Entry type="character" name="c">z</Entry>
            <Entry type="bytearray" name="blob">aaaa</Entry>
        </body>
    </message>
</messages>
```

Output: msg:type `map`

```
<?xml version="1.0" encoding="UTF-8"?>
<MapMessage>
    <Entry name="b" type="Byte">1</Entry>
    <Entry name="blob" encoding="base64" type="ByteArray">aaaa</Entry>
    <Entry name="s" type="Short">2</Entry>
    <Entry name="c" type="Character">z</Entry>
    <Entry name="bool" type="Boolean">true</Entry>
    <Entry name="d" type="Double">6.6</Entry>
    <Entry name="f" type="Float">5.5</Entry>
    <Entry name="firstKey" type="String">Blub</Entry>
    <Entry name="i" type="Integer">3</Entry>
    <Entry name="l" type="Long">4</Entry>
</MapMessage>
```

Input: msg:type `map` und headers_to_XML

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
    <headers>
        <header>
            <name>mySecondHeader</name>
            <value>the second</value>
        </header>
        <header>
            <name>JMSMessageID</name>
            <value>ID:4bedf9a6-05a1-11e7-bf59-eba4f4962570</value>
        </header>
        <header>
            <name>JMSXDeliveryCount</name>
            <value>1</value>
        </header>
        <header>
            <name>myFirstHeader</name>
            <value>the first</value>
        </header>
    </headers>
    <body>
        <MapMessage>
            <Entry name="b" type="Byte">1</Entry>
            <Entry name="blob" encoding="base64" type="ByteArray">aaaa</Entry>
            <Entry name="s" type="Short">2</Entry>
            <Entry name="c" type="Character">z</Entry>
            <Entry name="bool" type="Boolean">true</Entry>
            <Entry name="d" type="Double">6.6</Entry>
            <Entry name="f" type="Float">5.5</Entry>
            <Entry name="firstKey" type="String">Blub</Entry>
            <Entry name="i" type="Integer">3</Entry>
            <Entry name="l" type="Long">4</Entry>
        </MapMessage>
    </body>
</document>
```

10.14.4 Websphere MQ

10.14.4.1 IBM MQ Monitoring

? Unknown Attachment `IBM MQ Monitoring`: Überwacht einen *IBM Websphere MQ Server* mittels übergebener Kommandos.

> **Abhängigkeiten**
> Der Adapter benötigt die kostenpflichtige Bibliothek `com.ibm.mq.allclient.jar` im `Classpath`.

Eigenschaften

`Operation`	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`Query`*: Führt die im XML-Dokument übergebenen Kommandos aus und gibt diese in einem XML-Dokument aus

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.ibmmq.MQChannelChecker*: Hauptklasse (Standard)
MonitorIp	IP-Adresse des zu überwachenden MQ Servers **Mögliche Werte:** eine beliebige Zeichenkette (z. B. *`127.0.0.1`*)
MonitorPort	Port des zu überwachenden MQ Servers **Mögliche Werte:** eine beliebige positive Zahl (z. B. *`4711`*)
MonitorChannel	Name des Channels, der auf dem MQ Server überwacht werden soll **Mögliche Werte:** eine beliebige Zeichenkette (z. B. *`Channel`*)

Statuswerte

`1 (`*`successful`*`)`	Alle Befehle wurden erfolgreich durchgeführt
`-1 (`*`failed`*`)`	Die Anfrage schlug aufgrund eines technischen Fehlers fehl

Input

Dieser Adapter erwartet ein XML-Dokument als Input, das wie folgt aufgebaut ist:

```
<Commands>
    <{Command} />
</Commands>
```

Folgende Kommandos werden unterstützt:

- *`InquireQueue`*

- *InquireChannelNames*
- *InquireChannelStatus*

Beispiel für die drei Kommandos:

```
<Commands>
    <InquireQueue queue="{Wert}" />
    <InquireChannelNames channel="{Wert}" />
    <InquireChannelStatus channel="{Wert}" />
</Commands>
```

Output

Der Adapter erweitert im ausgegebenen XML-Dokument das Kommando-Element um das jeweilige Ergebnis.

Beispiel

```
<Commands>
    <InquireQueue queue="{Wert}">
        <Message>
            <Value name="{Wert}">{Wert}</Value>
        </Message>
    </InquireQueue>
    <InquireChannelNames channel="{Wert}">
        <NoResult />
    </InquireChannelNames>
    <InquireChannelStatus channel="{Wert}">
        <Message>
            <Value name="{Wert}">{Wert}</Value>
            <Value name="{Wert}">{Wert}</Value>
        </Message>
    </InquireChannelStatus>
</Commands>
```

-

10.14.4.2 MQ Series Transfer

`MQ Series Transfer`: Sendet und empfängt Nachrichten über eine *IBM Websphere MQ* (*MQ Series*) Nachrichten-Warteschlange und unterstützt MQ-spezifische Funktionen wie z. B. *Dynamic Queues*.

Ehemalige Bezeichnung bis Version 4.5.1: *MQ Series Transfer Adapter*

Eigenschaften (Properties)

Delete	(keine Funktion)
File name	(keine Funktion)
I/O Mode	(keine Funktion)
Operation	Bestimmt, welche Operation mit der Nachrichtenwarteschlange durchgeführt wird. **Mögliche Werte:** • *`Send`*: Daten aus dem vorangegangen Prozess-Schritt an die *MQ Series*-Warteschlange senden • *`Receive`*: Nachricht in den Prozess einlesen • *`Commit`*: Transaktion abschließen • *`Rollback`*: Transaktion abbrechen und sämtliche Änderungen rückgängig machen • *`Create Queue`*: Warteschlange erstellen • *`Destroy Queue`*: Warteschlange entfernen • `Clear`: Verbindung zu MQ Series zurücksetzen

Parameter

alias	Name der Parametergruppe zum externen Konfigurieren des Adapters über die Datei `QueueConfig.xml` (in `x4.ear` im Unterordner `conf`). Dort angegebene Werte besitzen eine höhere Priorität als die Parameter des Transfer-Adapter-Prozessbausteins.

toXML	Wenn Eigenschaft Operation auf *`Receive`* gesetzt wurde: Ergebnis des Prozess-Schritts als XML-Dokument ausgeben; Wenn zudem Parameter `MQMessage.format` auf *`NONE`* gesetzt wurde, dann wird die Nachricht Base64-kodiert **Mögliche Werte:** • *`true`*: XML-Dokument generieren, das Header und Body der Nachricht mit sämtlichen Eigenschaften enthält (siehe Beispiel) • *`false`*: Als Ergebnis des Prozess-Schritts eine MQ-Nachricht generieren (Standard)
ccdtUrl	URL zu einer *Client Channel Definition Table (CCDT)* ? Unknown Attachment Es gibt vier verschiedene Möglichkeiten, die `ccdtUrl` zu definieren: 1. Über die Java-Argumente `-DMQCHLLIB` und `-DMQCHLTAB` beim Start des *X4 Servers* **Mögliche Werte:** • `-DMQCHLLIB:` Pfad zur Table-Datei • `-DMQCHLTAB:` *`AMQCLCHL.TAB`* (IBM Default-Name) 2. Durch das Java-Argument `-Dx4.mqseries.ccdtUrl` Beispiel: `-Dx4.mqseries.ccdtUrl=file:///`*`Pfad_zur_Tabelle`*`/AMQCLCHL.TAB` 3. Über die Eigenschaft `ccdtUrl` in der *`QueueConfig.xml`* 4. Über den Parameter `ccdtUrl` im Adapter selbst Dabei wird die oben genannte Priorität angewendet.
RemoteQueueManager.name	Name des Queue-Managers **Mögliche Werte:** • Beliebige gültige Zeichenkette • *{EMPTY}*: Explizit kein Wert (leere Zeichenkette)
MQEnviroment.hostname	Hostname des angesprochenen MQ-Servers; gilt systemweit innerhalb der gesamten Java-VM für alle folgenden Operationen (der Parameter wird nur für lokale Tests auf Entwicklungsystemen benötigt, sonst leer lassen!) **Mögliche Werte:** • Beliebige Zeichenkette (z. B. *MQSR2402*) • *{EMPTY}*: Explizit kein Wert (leere Zeichenkette)

MQEnviroment.channel	Name des Channels;gilt systemweit innerhalb der gesamten Java-VM für alle folgenden Operationen (nur für lokale Tests auf Entwicklungsystemen benötigt, sonst leer lassen!) **Mögliche Werte:** • Beliebige Zeichenkette (z. B. *MQSR2402.CLS25740.DT*) • *{EMPTY}*: Explizit kein Wert (leere Zeichenkette)
MQEnviroment.CCSID	CCSID (Charset) aller Nachichten für die gesamte Java-VM; gilt systemweit innerhalb der gesamten Java-VM für alle folgenden Operationen, deshalb Vorsicht bei der Benutzung (sollte nicht benutzt werden!) **Mögliche Werte:** • Beliebige CCSID (z. B. *1208*) • *{EMPTY}*: Explizit kein Wert (leere Zeichenkette)
MQEnviroment.userID	Benutzer-ID zur Authentifizierung des *WebSphere MQ Clients* ; gilt systemweit innerhalb der gesamten Java-VM für alle folgenden Operationen (nur für lokale Tests auf Entwicklungsystemen benötigt, sonst leer lassen!)
MQEnviroment. localAddressSetting	MQ-spezifischer Parameter; gilt systemweit innerhalb der gesamten Java-VM für alle folgenden Operationen (nur für lokale Tests auf Entwicklungsystemen benötigt, sonst leer lassen!) **Mögliche Werte:** • Beliebige gültige Zeichenkette • *{EMPTY}*: Explizit kein Wert (leere Zeichenkette)
MQEnviroment.password	Passwort zur Authentifizierung des *WebSphere MQ Clients* ; gilt systemweit innerhalb der gesamten Java-VM für alle folgenden Operationen (nur für lokale Tests auf Entwicklungsystemen benötigt, sonst leer lassen!) **Mögliche Werte:** • Beliebige gültige Zeichenkette • *{EMPTY}*: Explizit kein Wert (leere Zeichenkette)

MQEnviroment.port	Port des MQServers; gilt systemweit innerhalb der gesamten Java-VM für alle folgenden Operationen (nur für lokale Tests auf Entwicklungsystemen benötigt, sonst leer) **Mögliche Werte:** • Beliebige ganze Zahl (z. B. *1414*) • *{EMPTY}*: Explizit kein Wert (leere Zeichenkette)
MQEnviroment.properties.MQC.SSL_CIPHER_SUITE_PROPERTY	`CipherSuite`-Konfiguration für die aktuelle Verbindung. Die eingetragene `CipherSuite` muss dabei der verwendeten `CipherSpec` entsprechen: **`CipherSpec` und die entsprechende `CipherSuite`** • *NULL_MD5*: *SSL_RSA_WITH_NULL_MD5* • *NULL_SHA*: *SSL_RSA_WITH_NULL_SHA* • *RC4_MD5_EXPORT*: *SSL_RSA_EXPORT_WITH_RC4_40_MD5* • *RC4_MD5_US*: *SSL_RSA_WITH_RC4_128_MD5* • *RC4_SHA_US*: *SSL_RSA_WITH_RC4_128_SHA* • *RC2_MD5_EXPORT*: *SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5* • *DES_SHA_EXPORT*: *SSL_RSA_WITH_DES_CBC_SHA* • *RC4_56_SHA_EXPORT1024*: *SSL_RSA_EXPORT1024_WITH_RC4_56_SHA* • *DES_SHA_EXPORT1024*: *SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA* • *TRIPLE_DES_SHA_US*: *SSL_RSA_WITH_3DES_EDE_CBC_SHA* • *TLS_RSA_WITH_AES_128_CBC_SHA*: *SSL_RSA_WITH_AES_128_CBC_SHA* • *TLS_RSA_WITH_AES_256_CBC_SHA*: S*SL_RSA_WITH_AES_256_CBC_SHA* • *AES_SHA_US* • *TLS_RSA_WITH_DES_CBC_SHA*: *SSL_RSA_WITH_DES_CBC_SHA* • *TLS_RSA_WITH_3DES_EDE_CBC_SHA*: *SSL_RSA_WITH_3DES_EDE_CBC_SHA* • *FIPS_WITH_DES_CBC_SHA*: *SSL_RSA_FIPS_WITH_DES_CBC_SHA* • *FIPS_WITH_3DES_EDE_CBC_SHA*: *SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA* ⓘ Zusätzliche Informationen zur `CipherSuite`-Konfiguration erhalten Sie auch hier (Kapitel 4) **Mögliche Werte:** `CipherSuite` als gültige Zeichenkette, s.o.

MQEnviroment.properties.MQC.TRANSPORT_PROPERTY	Einstellung für die aktuelle Verbindung **Mögliche Werte:** *MQC.TRANSPORT_MQSERIES*: Einziger zulässiger Wert
MQEnviroment.properties.MQC.CCSID_PROPERTY	CCSID (Charset) für die aktuelle Verbindung **Mögliche Werte:** Beliebige ganze Zahl (z. B. *1208*)
MQEnviroment.properties.MQC.USER_ID_PROPERTY	Benutzer-ID für die aktuelle Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQEnviroment.properties.MQC.CHANNEL_PROPERTY	Name des Channels für die aktuelle Verbindung **Mögliche Werte:** Beliebige Zeichenkette (z. B. *MQSR2402.CLS25740.DT*)
MQEnviroment.properties.MQC.CONNECT_ OPTIONS_PROPERTY	Verbindungsoptionen für die aktuelle Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQEnviroment.properties.MQC.HOST_NAME_PROPERTY	Hostname des angesprochenen MQ-Servers für die aktuelle Verbindung **Mögliche Werte:** Beliebige Zeichenkette (z. B. *MQSR2402*)
MQEnviroment.properties.MQC.LOCAL_ADDRESS_PROPERTY	Lokale Adresse für die aktuelle Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQEnviroment.properties.MQC.ORB_PROPERTY	ORB-Eigenschaft für die aktuelle Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQEnviroment.properties.MQC.PASSWORD_PROPERTY	Passwort-Eigenschaft für den aktuellen Verbindungsaufbau, um den *WebSphere MQ Client* zu authentifizieren (entspricht Parameter MQEnviroment.password) **Mögliche Werte:** Beliebige Zeichenkette
MQEnviroment.properties.MQC.PORT_PROPERTY	Port für die aktuelle Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQEnviroment.properties.MQC.RECEIVE_EXIT_PROPERTY	*Receive Exit* -Eigenschaft für die aktuelle Verbindung **Mögliche Werte:** Beliebige Zeichenkette

MQEnviroment.properties.MQC.SECURITY_EXIT_PROPERTY	*Security Exit* -Eigenschaft für die aktuelle Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQEnviroment.properties.MQC.SEND_EXIT_PROPERTY	*Send Exit* -Eigenschaft für die aktuelle Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQQueueManager.name	Name des QueueManagers. Nur für lokale Tests auf Entwicklungsystemen notwendig, sonst leer.
MQQueue.name	Name der Queue
MQQueue.dynamicQueueName	Name der Session Queue für diese Nachricht (Sollte zukünftig nicht mehr verwendet werden)
MQQueue.alternateUserId	Alternative User-ID um den Zugriff auf die Warteschlange zu prüfen, wenn diese geöffnet ist; dieser Parameter kann nicht gesetzt werden, während das Objekt geöffnet ist
MQQueue.openOptions.MQC.MQOO_OUTPUT	(Muss bei schreibendem Zugriff gesetzt werden)
MQQueue.openOptions.MQC.MQOO_FAIL_IF_QUIESCING	Erlaubt dem MQ-Manager, den Listener zu trennen (z. B. beim Herunterfahren)
MQQueue.openOptions.MQC.MQDL_SUPPORTED	MQ-spezifischer Parameter
MQQueue.openOptions.MQC.MQOO_ALTERNATE_USER_AUTHORITY	MQ-spezifischer Parameter
MQQueue.openOptions.MQC.MQOO_ BIND_AS_QDEF	Queue-Einstellungen zur Persistenz etc. übernehmen (sollte immer gesetzt werden) **Mögliche Werte:** • *ja*: Queue-Einstellungen übernehmen • *nein*: Keine Queue-Einstellungen übernehmen (Standard)
MQQueue_openOptions.MQC.MQOO_BIND_NOT_FIXED	MQ-spezifischer Parameter
MQQueue_openOptions.MQC.MQOO_BIND_ON_OPEN	MQ-spezifischer Parameter

MQQueue.openOptions.MQC.MQOO_INPUT_AS_Q_DEF	Einstellungen der Queue zur Persistenz etc. übernehmen (muss beim Lesen immer gesetzt werden) **Mögliche Werte:** • *ja*: Queue-Einstellungen übernehmen • *nein*: Keine Queue-Einstellungen übernehmen (Standard)
MQQueue.openOptions.MQC.MQOO_INPUT_SHARED	MQ-spezifischer Parameter
MQQueue.openOptions.MQC.MQOO_INPUT_EXCLUSIVE	MQ-spezifischer Parameter
MQQueue.openOptions.MQC.MQOO_INQUIRE	MQ-spezifischer Parameter
MQQueue.openOptions.MQC.MQOO_PASS_ALL_CONTEXT	MQ-spezifischer Parameter
MQQueue.openOptions.MQC.MQOO_PASS_IDENTITY_CONTEXT	MQ-spezifischer Parameter
MQQueue.openOptions.MQC.MQOO_SAVE_ALL_CONTEXT	MQ-spezifischer Parameter
MQQueue.openOptions.MQC.MQOO_SET	MQ-spezifischer Parameter
MQQueue.openOptions.MQC.MQOO_SET_ALL_CONTEXT	MQ-spezifischer Parameter
MQQueue.openOptions.MQC.MQOO_SET_IDENTITY_CONTEXT	MQ-spezifischer Parameter
MQPutMessageOptions.options.MQC.MQPMO_NEW_MSG_ID	MQ-spezifischer Parameter
MQPutMessageOptions.options.MQC.MQPMO_SYNCPOINT	Syncpoint beim Schreiben verwenden **Mögliche Werte:** • *ja*: Syncpoint beim Schreiben verwenden • *nein*: Keinen Syncpoint beim Schreiben verwenden (Standard)
MQQueue.closeOptions.MQC.MQCOO_DELETE	MQ-spezifischer Parameter

MQQueue.closeOptions.MQC.MQCOO_DELETE_PURGE	MQ-spezifischer Parameter
MQMessage.format	Nachrichtenformat **Mögliche Werte:** • *MQC.MQFMT_STRING*: Nachricht besteht komplett aus Zeichen • *MQC.MQFMT_ADMIN*: Befehl-Server-Anfrage/Antwort • *MQC.MQFMT_COMMAND1*: 1 Befehl-Server-Antwortnachricht vom Typ 1 • *MQC.MQFMT_COMMAND2*: 2 Befehl-Server-Antwortnachricht vom Typ2 • *MQC.MQFMT_DEAD_LETTER_HEADER*: Dead-letter Header • *MQC.MQFMT_EVENT*: Event • *MQC.MQFMT_NONE*: Kein Formatname, es findet keine Konvertierung statt (d. h. die Bytes werden original geliefert); wenn zudem Parameter toXML aktiviert ist, dann wird die Nachricht Base64-kodiert • *MQC.MQFMT_PCF*: Benutzerdefinierte Nachricht im *Programmable Command Format* • *MQC.MQFMT_TRIGGER*: Trigger-Nachricht • *MQC.MQFMT_XMIT_Q_HEADER*: Transmission Queue Header • *(kein Wert)*: Nicht konvertieren (entspricht MQC.MQFMT_NONE) (Standard)
MQMessage.applicationIdData	MQ-spezifischer Parameter
MQMessage.applicationOriginData	MQ-spezifischer Parameter
MQMessage.backoutCount	MQ-spezifischer Parameter
MQMessage.messageFlags.MQC.MQMT_REQUEST	MQ-spezifischer Parameter
MQMessage.replyToQueueName	Name der ReplyToQueue für diese Nachricht
MQMessage.replyToQueueManagerName	Name des ReplyToQueueManagers für diese Nachricht **Mögliche Werte:** Beliebige Zeichenkette

MQMessage.characterSet	Charset (CCSID) für die Nachricht; beim Auslesen: das erwartete Charset; Beim Schreiben: das Charset, das die Nachricht bekommt **Mögliche Werte:** Beliebige CCSID (z. B. *1208*)
MQMessage.correlationId	CorrelationID mit 24 Byte Länge (diese ist ebenso wie die zurückgegebene MessageID Base64-kodiert) **Mögliche Werte:** Beliebige Zeichenkette mit 32 Zeichen, die eine Base64-kodierte CorrelationID mit 24 Byte Länge enthält
MQMessage.encoding	Zeichenkodierung der Nachricht
MQMessage.expiry	Zeit in Zehntelsekunden, nach der die Nachricht ungültig wird und aus der Queue gelöscht wird. **Mögliche Werte:** Beliebige ganze Zahl
MQMessage.feedback	MQ-spezifischer Parameter
MQMessage.groupId	MQ-spezifischer Parameter
MQMessage.messageId	MQ-spezifischer Parameter
MQMessage.messageSequenceNumber	MQ-spezifischer Parameter
MQMessage.messageType	MQ-spezifischer Parameter
MQMessage.offset	MQ-spezifischer Parameter
MQMessage.originalLength	MQ-spezifischer Parameter
MQMessage.persistence	MQ-spezifischer Parameter
MQMessage.priority	MQ-spezifischer Parameter
MQMessage.putApplicationType	MQ-spezifischer Parameter
MQMessage.report	MQ-spezifischer Parameter
MQMessage.userId	MQ-spezifischer Parameter

MQGetMessageOptions.options.MQC.MQGMO_SYNCPOINT	Syncpoint beim Empfangen von Nachrichten verwenden (erforderlich für die Operationen Commit und Rollback) Mögliche Werte: • *ja*: Syncpoint beim Empfangen verwenden • *nein*: Keinen Syncpoint beim Empfangen verwenden (Standard)
MQGetMessageOptions.options.MQC.MQGMO_WAIT	Beim Auslesen warten, wenn sich keine Nachricht in der Queue befindet
MQGetMessageOptions.options.MQC.MQGMO_NO_WAIT	Beim Auslesen nicht warten, wenn sich keine Nachricht in der Queue befindet, und sofort abbrechen **Mögliche Werte:** • *ja*: Nicht warten und sofort abbrechen • *nein*: Warten (Standard)
MQGetMessageOptions.options.MQC.MQGMO_ACCEPT_TRUNCATED_MSG	MQ-spezifischer Parameter
MQGetMessageOptions.options.MQC.MQGMO_BROWSE_FIRST	MQ-spezifischer Parameter
MQGetMessageOptions.options.MQC.MQGMO_BROWSE_MSG_UNDER_CURSOR	MQ-spezifischer Parameter
MQGetMessageOptions.options.MQC.MQGMO_BROWSE_NEXT	MQ-spezifischer Parameter
MQGetMessageOptions.options.MQC.MQGMO_CONVERT	Automatische Konvertierung in MQ aktivieren **Mögliche Werte:** • *ja*: Konvertierung aktivieren • *nein*: Keine Konvertierung (Standard)
MQGetMessageOptions.options.MQC.MQGMO_FAIL_IF_QUIESCING	MQ-spezifischer Parameter
MQGetMessageOptions.options.MQC.MQGMO_LOCK	MQ-spezifischer Parameter

MQGetMessageOptions.options.MQC.MQGMO_MSG_UNDER_CURSOR	MQ-spezifischer Parameter
MQGetMessageOptions.options.MQC.MQGMO_NONE	Alle anderen *GetOptions* ignorieren **Mögliche Werte:** • *ja*: Alle anderen GetOptions ignorieren • *nein*: Andere GetOptions nicht ignorieren (Standard)
MQGetMessageOptions.options.MQC.MQGMO_MARK_SKIP_BACKOUT	MQ-spezifischer Parameter
MQGetMessageOptions.options.MQC.MQGMO_NO_SYNCPOINT	Syncpoint deaktivieren **Mögliche Werte:** • *ja*: Syncpoint deaktivieren (damit ist keine Transaktion möglich) • *nein*: Syncpoint nicht deaktivieren (Standard)
MQGetMessageOptions.options.MQC.MQGMO_SYNCPOINT_IF_PERSISTENT	Syncpoint nutzen, wenn die Nachricht als persistent gekennzeichnet ist **Mögliche Werte:** • *ja*: Syncpoint bei persistenten Nachrichten nutzen • *nein*: Keinen Syncpoint bei persistenten Nachrichten nutzen (Standard)
MQGetMessageOptions.options.MQC.MQGMO_UNLOCK	MQ-spezifischer Parameter
MQGetMessageOptions.options.MQC.MQGMO_ALL_MSGS_AVAILABLE	MQ-spezifischer Parameter
MQGetMessageOptions.options.MQC.MQGMO_ALL_SEGMENTS_AVAILABLE	MQ-spezifischer Parameter
MQGetMessageOptions.options.MQC.MQGMO_COMPLETE_MSG	MQ-spezifischer Parameter

MQGetMessageOptions.options.MQC.MQGMO_LOGICAL_ORDER	MQ-spezifischer Parameter
MQGetMessageOptions.matchOptions.MQC.MQGMO_MATCH_CORREL_ID	MQ-spezifischer Parameter
MQGetMessageOptions.matchOptions.MQC.MQGMO_MATCHGROUP_ID	MQ-spezifischer Parameter
MQGetMessageOptions.matchOptions.MQC.MQGMO_MATCH_MSG_ID	MQ-spezifischer Parameter
MQGetMessageOptions.matchOptions.MQC.MQGMO_MATCH_MSG_SEQ_NUMBER	MQ-spezifischer Parameter
MQGetMessageOptions.matchOptions.MQC_MQGMO_NONE	MQ-spezifischer Parameter
MQGetMessageOptions.waitInterval	Zeit in Millisekunden, die beim Auslesen gewartet wird, wenn sich keine Nachrichten in der Queue befinden (siehe Parameter MQGetMessageOptions.options.MQC.MQGMO_WAIT). Der Nachrichtenempfang wird nicht abgebrochen, wenn er während des Warte-Intervalls begonnen wurde. **Mögliche Werte:** Beliebige positive ganze Zahl
MQGetMessageOptions.resolvedQueueName	MQ-spezifischer Parameter

Weitere Informationen zu den MQ-spezifischen Parametern finden Sie unter https://www.ibm.com/docs/en/ibm-mq/8.0?topic=java-mqenvironment.

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Beispiel

Das erzeugte XML-Dokument, wenn Eigenschaft `Operation` auf *`Receive`* gesetzt wurde, besitzt folgende Struktur:

```
<document>
<header>
   <CorrelationID>AAAAAAAAAAAAAAAAAAAA</CorrelationID>
   <MessageID>QU1RIFFNX3NwX2Nvbm5lYxvAdUQgC9sH</MessageID>
   <MessageSequenceNr>1</MessageSequenceNr>
   <MessageFormat>MQSTR</MessageFormat>
   <ReplyToQueue/>
   <ReplyToQueueMngr>QM_sp_connect</ReplyToQueueMngr>
   <backoutCount>0</backoutCount>
</header>
<body>Message</body>
</document>
```

Weitere Informationen zu den *MQ Series*-spezifischen Parametern erhalten Sie von dessen Hersteller *IBM* in der englischsprachigen Dokumentation *WebSphere MQ – Using Java* In Kapitel *9 The WebSphere MQ base Java classes and interfaces*.

10.14.4.3 Websphere MQ Connector

`Websphere MQ Connector`: Sendet, empfängt und zählt Nachrichten einer *IBM Websphere MQ*-Nachrichten-Warteschlange und unterstützt dabei die *MQ*-Version 7.

Ehemalige Bezeichnung bis Version 4.5.1: *IBM Websphere MQ Adapter*

Beachten Sie, dass dieser Adapter unabhängig von der Warteschlangen-Konfiguration des *MQ Series Transfer Adapters* (innerhalb von `x4.ear` erfolgt dies in der Datei `conf/QueueConfig.xml`) arbeitet.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation mit der Nachrichtenwarteschlange durchgeführt wird. Mögliche Werte: • *Send*: Daten aus dem vorangegangen Prozess-Schritt an die *MQ Series*-Warteschlange senden • *Receive*: Nachricht in den Technical Process einlesen • *Commit*: Transaktion abschließen • *Rollback*: Transaktion abbrechen und sämtliche Änderungen rückgängig machen • *Count*: Anzahl der Nachrichten in der Warteschlange in einem XML-Dokument ausgeben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.ibmmq.MQAdapter*: Hauptklasse (Standard)
toXML	Wenn Eigenschaft Operation auf *Receive* gesetzt wurde: Ergebnis des Prozess-Schritts als XML-Dokument ausgeben; Wenn zudem Parameter `MQMessage.format` auf *`NONE`* gesetzt wurde, dann wird die Nachricht Base64-kodiert **Mögliche Werte:** • *`true`*: XML-Dokument generieren, das Header und Body der Nachricht mit sämtlichen Eigenschaften enthält (siehe Beispiel) • *`false`*: Als Ergebnis des Prozess-Schritts eine MQ-Nachricht generieren (Standard)
ccdtUrl	URL zu einer *Client Channel Definition Table (CCDT)* Es gibt drei verschiedene Möglichkeiten, die `ccdtUrl` zu definieren: 1. Über die Java-Argumente `-DMQCHLLIB` und `-DMQCHLTAB` beim Start des *X4 Servers* **Mögliche Werte:** `-DMQCHLLIB`:Pfad zur *Table*-Datei `-DMQCHLTAB`: *AMQCLCHL.TAB* (IBM Default-Name) 2. Durch das Java-Argument *`-Dx4.mqseries.ccdtUrl`* *Beispiel:* `-Dx4.mqseries.ccdtUrl=file:///`*`Pfad_zur_Tabelle`*`/AMQCLCHL.TAB` 3. Über den Parameter `ccdtUrl` im Adapter selbst Dabei wird die oben genannte Priorität angewendet.
RemoteQueueManager_name	Name des Queue-Managers
MQEnvironment_hostname	Hostname des angesprochenen MQ-Servers; gilt Java-VM-weit (der Parameter wird nur für lokale Tests auf Entwicklungsystemen benötigt, sonst bleibt er leer) **Mögliche Werte:** Beliebige Zeichenkette (z.B. *MQSR2402*)

MQEnvironment_ channel	Name des Channels; gilt Java-VM-weit (nur für lokale Tests auf Entwicklungsystemen benötigt, sonst leer) **Mögliche Werte:** Beliebige Zeichenkette (z.B. *MQSR2402.CLS25740.DT*)
MQEnvironment_CCSID	CCSID (Charset) aller Nachichten für die gesamte Java-VM; deshalb Vorsicht bei der Benutzung (sollte nicht benutzt werden!) **Mögliche Werte:** Beliebige CCSID (z.B. *1208*)
MQEnvironment_userID	Benutzer-ID zur Authentifizierung des *WebSphere MQ Clients*
MQEnvironment_ localAddress Setting	MQ-spezifischer Parameter
MQEnvironment_ password	Passwort zur Authentifizierung des *WebSphere MQ Clients*
MQEnvironment_port	Port des MQ-Servers; gilt Java-VM-weit (nur für lokale Tests auf Entwicklungsystemen benötigt, sonst leer) **Mögliche Werte:** Beliebige ganze Zahl (z.B. *1414*)
MQEnvironment_ properties_MQC_ TRANSPORT_PROPERTY	Einstellung für die Verbindung **Mögliche Werte:** *MQC.TRANSPORT_MQSERIES*: Einziger zulässiger Wert
MQEnvironment_ properties_MQC_ CCSID_PROPERTY	CCSID (Charset) für die Verbindung **Mögliche Werte:** Beliebige ganze Zahl (z.B. *1208*)
MQEnvironment_ properties_MQC_ USER_ID_PROPERTY	Benutzer-ID für die Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQEnvironment_ properties_MQC_ CHANNEL_PROPERTY	Name des Kanals für die Verbindung **Mögliche Werte:** Beliebige Zeichenkette (z.B. *MQSR2402.CLS25740.DT*)
MQEnvironment_ properties_MQC_ CONNECT_OPTIONS_PROPERTY	Verbindungsoptionen **Mögliche Werte:** Beliebige Zeichenkette

MQEnvironment_ properties_MQC_ HOST_NAME_PROPERTY	Hostname des angesprochenen MQ-Servers für die Verbindung **Mögliche Werte:** Beliebige Zeichenkette (z.B. *MQSR2402*)
MQEnvironment_ properties_MQC_ LOCAL__ADDRESS_PROPERTY	Lokale Adresse für die Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQEnvironment_ properties_MQC_ ORB_PROPERTY	ORB-Eigenschaft für die Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQEnvironment_ properties_MQC_ PASSWORD_PROPERTY	Passwort-Eigenschaft für den Verbindungsaufbau, um den *WebSphere MQ Client* zu authentifizieren (entspricht Parameter MQEnvironment_password) **Mögliche Werte:** Beliebige Zeichenkette
MQEnvironment_ properties_MQC_ PORT_PROPERTY	Port für die Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQEnvironment_ properties_MQC_ RECEIVE_EXIT_PROPERTY	*Receive Exit* -Eigenschaft für die Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQEnvironment_ properties_MQC_ SECURITY_EXIT_PROPERTY	*Security Exit* -Eigenschaft für die Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQEnvironment_ properties_MQC_ SEND_EXIT_PROPERTY	*Send Exit* -Eigenschaft für die Verbindung **Mögliche Werte:** Beliebige Zeichenkette
MQQueueManager_name	Name des QueueManagers (nur für lokale Tests auf Entwicklungsystemen notwendig, sonst leer)
MQQueue_name	Name der Queue **Mögliche Werte:** Beliebige gültige Zeichenkette
MQQueue_dynamicQueue Name	Name der *Session Queue* für diese Nachricht (Sollte zukünftig nicht mehr verwendet werden) **Mögliche Werte:** Beliebige gültige Zeichenkette

MQQueue_alternateUserId	Alternative User-ID um den Zugriff auf die Warteschlange zu prüfen, wenn diese geöffnet ist; dieser Parameter kann nicht gesetzt werden, während das Objekt geöffnet ist
MQQueue_openOptions_ MQC_MQOO_OUTPUT	(Muss bei schreibendem Zugriff gesetzt werden)
MQQueue_openOptions_ MQC_MQOO_FAIL_IF_ QUIESCING	Erlaubt dem MQ-Manager, den Listener zu trennen (z.B. beim Herunterfahren)
MQQueue_openOptions_ MQC_MQDL_SUPPORTED	MQ-spezifischer Parameter
MQQueue_openOptions_ MQC_MQOO_ALTERNATE_ USER_AUTHORITY	MQ-spezifischer Parameter
MQQueue_openOptions_ MQC_MQOO_BIND_AS_QDEF	Queue-Einstellungen zur Persistenz etc. übernehmen (sollte immer gesetzt werden) **Mögliche Werte:** • `true`: Queue-Einstellungen übernehmen • `false`: Keine Queue-Einstellungen übernehmen (Standard)
MQQueue_openOptions_ MQC_MQOO_BIND_NOT_ FIXED	MQ-spezifischer Parameter
MQQueue_openOptions_ MQC_MQOO_BIND_ON_OPEN	MQ-spezifischer Parameter
MQQueue_openOptions_ MQC_MQOO_INPUT_AS_Q_DEF	Einstellungen der Queue zur Persistenz etc. übernehmen (Muss beim Lesen immer gesetzt werden) **Mögliche Werte:** • *`true`*: Queue-Einstellungen übernehmen • *`false`*: Keine Queue-Einstellungen übernehmen (Standard)
MQQueue_openOptions_ MQC_MQOO_INPUT_SHARED	MQ-spezifischer Parameter
MQQueue_openOptions_ MQC_MQOO_INPUT_ EXCLUSIVE	MQ-spezifischer Parameter
MQQueue_openOptions_ MQC_MQOO_INQUIRE	MQ-spezifischer Parameter
MQQueue_openOptions_ MQC_MQOO_PASS_ALL_ CONTEXT	MQ-spezifischer Parameter

MQQueue_openOptions_ MQC_MQOO_PASS_ IDENTITY_CONTEXT	MQ-spezifischer Parameter
MQQueue_openOptions_ MQC_MQOO_SAVE_ALL_ CONTEXT	MQ-spezifischer Parameter
MQQueue_openOptions_ MQC_MQOO_SET	MQ-spezifischer Parameter
MQQueue_openOptions_ MQC_MQOO_SET_ALL_CONTEXT	MQ-spezifischer Parameter
MQQueue_openOptions_ MQC_MQOO_SET_IDENTITY_ CONTEXT	MQ-spezifischer Parameter
MQPutMessageOptions_ options_MQC_MQPMO_ NEW_MSG_ID	MQ-spezifischer Parameter
MQPutMessageOptions_ options_MQC_MQPMO_ SYNCPOINT	Syncpoint beim Schreiben von Nachrichten verwenden (erforderlich für die Operationen Commit und Rollback) **Mögliche Werte:** • *true*: Syncpoint beim Schreiben verwenden • *false*: Keinen Syncpoint beim Schreiben verwenden (Standard)
MQQueue_closeOptions_ MQC_MQCOO_DELETE	MQ-spezifischer Parameter
MQQueue_closeOptions_ MQC_MQCOO_DELETE_PURGE	MQ-spezifischer Parameter

MQMessage_format	Nachrichtenformat **Mögliche Werte:** • *MQC.MQFMT_STRING*: Nachricht besteht komplett aus Zeichen • *MQC.MQFMT_ADMIN*: Befehl-Server-Anfrage/Antwort • *MQC.MQFMT_COMMAND1*: 1 Befehl-Server-Antwortnachricht vom Typ 1 • *MQC.MQFMT_COMMAND2*: 2 Befehl-Server-Antwortnachricht vom Typ2 • *MQC.MQFMT_DEAD_LETTER_HEADER*: Dead-letter Header • *MQC.MQFMT_EVENT*: Event • *MQC.MQFMT_NONE*: Kein Formatname, es findet keine Konvertierung statt (d. h. die Bytes werden original geliefert); wenn zudem Parameter toXML aktiviert ist, dann wird die Nachricht Base64-kodiert • *MQC.MQFMT_PCF*: Benutzerdefinierte Nachricht im *Programmable Command Format* • *MQC.MQFMT_TRIGGER*: Trigger-Nachricht • *MQC.MQFMT_XMIT_Q_HEADER*: Transmission Queue Header • *(kein Wert)*: Nicht konvertieren (entspricht MQC.MQFMT_NONE) (Standard)
MQMessage_applicationIdData	MQ-spezifischer Parameter
MQMessage_applicationOriginData	MQ-spezifischer Parameter
MQMessage_backoutCount	MQ-spezifischer Parameter
MQMessage_messageFlags_MQC_MQMT_REQUEST	MQ-spezifischer Parameter
MQMessage_replyToQueueName	Name der ReplyToQueue für diese Nachricht
MQMessage_replyToQueueManagerName	Name des ReplyToQueueManagers für diese Nachricht **Mögliche Werte:** Beliebige Zeichenkette

MQMessage_characterSet	Charset (CCSID) für die Nachricht; beim Auslesen: das erwartete Charset; Beim Schreiben: das Charset, das die Nachricht bekommt **Mögliche Werte:** Beliebige CCSID (z.B. *1208*)
MQMessage_correlationId	CorrelationID mit 24 Byte Länge (diese ist ebenso wie die zurückgegebene MessageID Base64-kodiert) **Mögliche Werte:** Beliebige Zeichenkette mit 32 Zeichen, die eine Base64-kodierte CorrelationID mit 24 Byte Länge enthält
MQMessage_encoding	Zeichenkodierung der Nachricht
MQMessage_expiry	Zeit in Zehntelsekunden, nach der die Nachricht ungültig wird und aus der Queue gelöscht wird **Mögliche Werte:** Beliebige ganze Zahl
MQMessage_feedback	MQ-spezifischer Parameter
MQMessage_groupId	MQ-spezifischer Parameter
MQMessage_message Id	MQ-spezifischer Parameter
MQMessage_message SequenceNumber	MQ-spezifischer Parameter
MQMessage_messageType	MQ-spezifischer Parameter
MQMessage_offset	MQ-spezifischer Parameter
MQMessage_originalLength	MQ-spezifischer Parameter
MQMessage_persistence	MQ-spezifischer Parameter
MQMessage_priority	MQ-spezifischer Parameter
MQMessage_put ApplicationType	MQ-spezifischer Parameter
MQMessage_report	MQ-spezifischer Parameter
MQMessage_userId	MQ-spezifischer Parameter

MQGetMessageOptions_ options_MQC_MQGMO_ SYNCPOINT	Syncpoint beim Empfangen von Nachrichten verwenden (erforderlich für die Operationen Commit und Rollback) **Mögliche Werte:** • `true`: Syncpoint beim Empfangen verwenden • `false`: Keinen Syncpoint beim Empfangen verwenden (Standard)
MQGetMessageOptions_ options_MQC_MQGMO_WAIT	Beim Auslesen warten, wenn sich keine Nachricht in der Queue befindet
MQGetMessageOptions_ options_MQC_MQGMO_ NO_WAIT	Beim Auslesen nicht warten, wenn sich keine Nachricht in der Queue befindet, und sofort abbrechen **Mögliche Werte:** • `true`: Nicht warten und sofort abbrechen • `false`: Warten (Standard)
MQGetMessageOptions_ options_ MQC_MQGMO_ ACCEPT_TRUNCATED_MSG	MQ-spezifischer Parameter
MQGetMessageOptions_ options_MQC_MQGMO_ BROWSE_FIRST	MQ-spezifischer Parameter
MQGetMessageOptions_ options_MQC_MQGMO_ BROWSE_MSG_UNDER_CURSOR	MQ-spezifischer Parameter
MQGetMessageOptions_ options_MQC_MQGMO_ BROWSE_NEXT	MQ-spezifischer Parameter
MQGetMessageOptions_ options_MQC_MQGMO_ CONVERT	Automatische Konvertierung in MQ aktivieren **Mögliche Werte:** • `true`: Konvertierung aktivieren • `false`: Keine Konvertierung (Standard)
MQGetMessageOptions_ options_MQC_MQGMO_FAIL_ IF_QUIESCING	MQ-spezifischer Parameter
MQGetMessageOptions_ options_MQC_MQGMO_LOCK	MQ-spezifischer Parameter
MQGetMessageOptions_ options_MQC_MQGMO_ MSG_UNDER_CURSOR	MQ-spezifischer Parameter

MQGetMessageOptions_ options_MQC_MQGMO_NONE	Alle anderen *GetOptions* ignorieren **Mögliche Werte:** • `true`: Alle anderen GetOptions ignorieren • `false`: Andere GetOptions nicht ignorieren (Standard)
MQGetMessageOptions_ options_MQC_MQGMO_ MARK_SKIP_BACKOUT	MQ-spezifischer Parameter
MQGetMessageOptions_ options_MQC_ MQGMO_NO_SYNCPOINT	Syncpoint deaktivieren **Mögliche Werte:** • `true`: Syncpoint deaktivieren (damit ist keine Transaktion möglich) • `false`: Syncpoint nicht deaktivieren (Standard)
MQGetMessageOptions_ options_MQC_MQGMO_ SYNCPOINT_IF_PERSISTENT	Syncpoint nutzen, wenn die Nachricht als persistent gekennzeichnet ist **Mögliche Werte:** • `true`: Syncpoint bei persistenten Nachrichten nutzen • `false`: Keinen Syncpoint bei persistenten Nachrichten nutzen (Standard)
MQGetMessageOptions_ options_MQC_MQGMO_ UNLOCK	MQ-spezifischer Parameter
MQGetMessageOptions_ options_MQC_MQGMO_ ALL_MSGS_AVAILABLE	MQ-spezifischer Parameter
MQGetMessageOptions_ options_MQC_MQGMO_ALL_ SEGMENTS_AVAILABLE	MQ-spezifischer Parameter
MQGetMessageOptions_ options_MQC_ MQGMO_COMPLETE_MSG	MQ-spezifischer Parameter
MQGetMessageOptions_ options_MQC_ MQGMO_LOGICAL_ORDER	MQ-spezifischer Parameter
MQGetMessageOptions_ matchOptions_MQC_ MQGMO_MATCH_CORREL_ID	MQ-spezifischer Parameter
MQGetMessageOptions_ matchOptions_MQC_ MQGMO_MATCH_GROUP_ID	MQ-spezifischer Parameter

MQGetMessageOptions_ matchOptions_MQC_ MQGMO_MATCH_MSG_ID	MQ-spezifischer Parameter
MQGetMessageOptions_ matchOptions_MQC_ MQGMO_MATCH_MSG_ SEQ_NUMBER	MQ-spezifischer Parameter
MQGetMessageOptions_ matchOptions_ MQC_MQGMO_NONE	MQ-spezifischer Parameter
MQGetMessageOptions_ waitInterval	Zeit in Millisekunden, die beim Auslesen gewartet wird, wenn sich keine Nachrichten in der Queue befinden (siehe Parameter `MQGetMessageOptions_options_MQC_MQGMO_WAIT`). Der Nachrichtenempfang wird nicht abgebrochen, wenn er während des Warte-Intervalls begonnen wurde. **Mögliche Werte:** Beliebige positive ganze Zahl
MQGetMessageOptions_ resolvedQueueName	MQ-spezifischer Parameter

Weitere Informationen zu den *MQ Series* spezifischen Parametern erhalten Sie von dessen Hersteller *IBM* in der englischsprachigen Dokumentation *WebSphere MQ – Using Java* In Kapitel *9 The WebSphere MQ base Java classes and interfaces.*

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Operation: Receive

Mit Operation `Receive` lassen sich Nachrichten aus der Warteschlange in den Technical Process einlesen.

Input
Für Operation `Receive` erwartet der Adapter kein Input-Dokument.

Ausgabe
Für Operation `Receive` und wenn Eigenschaft `Operation` auf *`Receive`* gesetzt wurde, gibt der Adapter beispielsweise folgendes XML-Dokument aus:

```
<document>
<header>
   <CorrelationID>AAAAAAAAAAAAAAAAAAAA</CorrelationID>
   <MessageID>QU1RIFFNX3NwX2Nvbm5lYxvAdUQgC9sH</MessageID>
   <MessageSequenceNr>1</MessageSequenceNr>
   <MessageFormat>MQSTR</MessageFormat>
   <ReplyToQueue/>
   <ReplyToQueueMngr>QM_sp_connect</ReplyToQueueMngr>
   <backoutCount>0</backoutCount>
</header>
<body>Message</body>
</document>
```

Operation: Count

Input
Für Operation `Count` erwartet der Adapter kein Input-Dokument.

Ausgabe
Für Operation `Count` gibt der Adapter ein XML-Dokument aus, das im Wurzel-Element `<Count>` die Anzahl der Nachrichten in der Warteschlange enthält.

```
<Count><!--Nachrichtenanzahl--><Count>
```

10.15 Office

- Microsoft Excel
- Microsoft Word Properties
- Microsoft Word to Text Converter
- Working Time Calculator

10.15.1 Microsoft Excel

10.15.1.1 Microsoft Excel Reader

`Microsoft Excel Reader`: Liest ein Microsoft-Excel-Dokument ein (im `.xls`-Format von Excel 97–2003 oder im XML-basierten OOXML-Format `.xlsx`) und gibt dessen Inhalt als XML-Dokument aus.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Funktionsadapter durchführt. **Mögliche Werte:** • *`Read Spreadsheet`*: Excel-Dokument lesen

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** • *de.softproject.integration.adapter.excel.ExcelReader*: Hauptklasse (Standard)
`Formatter`	Verwendeter Formatierer (hat Einfluss auf die XML-Ausgabe, siehe unten) **Mögliche Werte:** • *`de.softproject.integration.adapter.excel.DefaultFormatter`*: Jeden Zelleninhalt des Excel-Dokuments in einem Element `<Cell/>` ausgeben (Standard) • *`de.softproject.integration.adapter.excel.NumberedCellFormatter`*: Jeden Zelleninhalt der Excel-Datei in einem XML-Element mit Zellennummer ausgeben (z. B. *`<Cell_0/>`*) • *`de.softproject.integration.adapter.excel.LabeledCellFormatter`*: Jeden Zelleninhalt der Excel-Datei in einem XML-Element mit Zellenbuchstaben ausgeben (z. B. *`<Cell_A/>`*)

`MinimumInflateRatio`	Optional. Gibt das minimale Dekompressionsverhältnis an, das beim Lesen einer Datei zulässig ist. In Apache POI wurde eine Sicherheitsprüfung eingeführt, um Angriffe durch sogenannte ZIP-Bomben abzufangen. Bei diesen Angriffen steigt die Größe einer Datei beim Entpacken in den Arbeitsspeicher um ein Vielfaches an. Das kann problematisch sein, wenn der Arbeitsspeicher der Java Virtual Machine mit der Größe dekomprimierten Datei überfordert ist, da es dadurch zu einem Serverausfall kommen kann. Die eingeführte Sicherheitsprüfung verwendet ein minimales Dekompressionsverhältnis. Dieser Wert misst die Differenz zwischen dem von einer Datei im Dateisystem beanspruchten Speicherplatz und dem beim Lesen der Datei benötigten Arbeitsspeicher. **Mögliche Werte:** • Beliebige ganze Zahl zwischen 0 und 100 • Der Standardwert ist 1. Mit diesem Wert schlägt der Adapter z. B. dann fehl, wenn eine Datei im Dateisystem eine Größe von 1 KB aufweist, im Arbeitsspeicher jedoch eine Größe von 100 KB. • Mit dem Wert 0 wird die Sicherheitsprüfung deaktiviert. • Bei einem Wert von 100 liest der Adapter keines der bereitgestellten Dokumente aus, da eine Datei in einem Fall im Arbeitsspeicher kleiner ist als im Dateisystem. • Wenn der angegebene Wert nicht im zulässigen Bereich (zwischen 0 und 100) liegt oder ungültig ist, wird der Standardwert verwendet

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Output

Mit dem Standardformatierer `de.softproject.integration.adapter.excel.DefaultFormatter` wird folgende XML-Struktur ausgegeben:

```
<Workbook>
   <Sheet name="Name des Tabellenblatts">
      <Row rowId="Zeilennummer">
         <Cell colId="Spalten-ID" type="Datentyp" format="Formatierung"
               formula="Formel"><!--Zelleninhalt--></Cell>
      </Row>
   </Sheet>
</Workbook>
```

Mit dem Formatierer `de.softproject.integration.adapter.excelWriter.NumberedCellFormatter` wird folgende XML-Struktur ausgegeben:

```
<Workbook>
   <Sheet name="Name des Tabellenblatts">
      <Row rowId="Zeilennummer">
         <Cell_0 colId="Zellen-ID" type="Datentyp" format="Formatierung"
                 formula="Formel"><!--Zelleninhalt--></Cell_0>
      </Row>
   </Sheet>
</Workbook>
```

Mit dem Formatierer `de.softproject.integration.adapter.excel.LabeledCellFormatter` wird folgende XML-Struktur ausgegeben:

```
<Workbook>
   <Sheet name="Name des Tabellenblatts">
      <Row rowId="Zeilennummer">
         <Cell_A colId="Zellen-ID" type="Datentyp" format="Formatierung"
                 formula="Formel"><!--Zelleninhalt--></Cell_A>
      </Row>
   </Sheet>
</Workbook>
```

10.15.1.2 Microsoft Excel Writer

`Microsoft Excel Writer`: Konvertiert ein XML-Dokument mit einer bestimmten Tabellenstruktur in eine *Microsoft Excel* -Datei (im `.xls`-Format *Excel 97 – 2003* oder im XML-basierten OOXML-Format `.xlsx`) oder füllt eine bestehende Exceldatei (im `.xls`-Format *Excel 97 – 2003* oder im XML-basierten *OOXML* -Format `.xlsx`) mit Daten aus einem XML-Dokument mit einer bestimmten Tabellenstruktur.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • `Create Spreadsheet`: Excel-Dokument mit einem Tabellenblatt ausgeben, die den MIME-Typ *`Application/Vnd.ms-excel`* besitzt • `Fill Template`: Bestehende Excel-Vorlage, die in Parameter `TemplateURL` angegeben wird und beliebig viele Tabellenblätter besitzen kann, mit Daten füllen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.excel.ExcelWriter*: Hauptklasse (Standard)
generateOOXML	Excel-Dateiformat (Microsoft Excel 97 – 2003 oder OOXML) bestimmen **Mögliche Werte:** • *`true`*: XML-basiertes Excel-Dokument (Office Open XML; .xlsx) erzeugen • *`false`*: Excel-Dokument im Format Microsoft Excel 97 – 2003 (.xls) erzeugen (Standard)
TemplateUrl	URL zur Excel-Vorlage (wenn die Eigenschaft Operation auf *`Fill Template`* gesetzt ist). Je nach dem in Parameter *`generateOOXML`* gewählten Dateiformat, muss die angegebene Vorlage in demselben Format vorliegen; z.B. muss eine `.xlsx`-Vorlage verwendet werden, wenn der Parameter *`generateOOXML`* gesetzt wurde. Der Wert des Parameters `TemplateUrl` kann durch das Attribut *`templateUrl=".."`* im Input-XML überschrieben werden (siehe unten). **Mögliche Werte:** • URL mit file://-Protokollangabe (z.B. *`file:///%USER_COLLECTION%/Project/Template.xls`*) • URL mit xstore://-Protokollangabe (z.B. *`xstore://Project/Folder/Template.xlsx`*)

recalc	Formeln vor dem Speichern des Dokuments neu berechnen und Feldwerte aktualisieren **Mögliche Werte:** • *true*: Formeln neu berechnen • *false*: Werte von Feldern mit Formeln unverändert lassen (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Je nach gewählter Operation erwartet der Adapter eine spezifische Input-Struktur:

- **Operation `Create Spreadsheet`:**
  Für diese Operation erwartet der Adapter folgende XML-Struktur, die dem Aufbau von (X)HTML-Tabellen entspricht. Hierbei erzeugt das Element `<table>` ein Tabellenblatt, jedes Element `<tr>` eine neue Tabellenzeile und jedes Element `<td>` eine neue Zelle oder `<th>` eine Kopfzelle. Wenn mehrere Tabellenblätter erzeugt werden sollen, müssen die `<table>`-Elemente von einem Wurzel-Element mit beliebigem Namen umschlossen werden.

  > Der Name der Tabelle, der über das Attribut `name` im Element `<table>` festgelegt wird, darf maximal 31 Zeichen lang sein.

  > Die Formatierung der einzelnen Zellen lässt sich über verschiedene Attribute steuern. Diese werden im Abschnitt **Mögliche Attribute** erläutert.

Beispiel-Input für die Operation Create Spreadsheet

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
 <table name="Name of the first spreadsheet">
   <tr autofit="true" height="10">
     <th fontHeight="12" forceString="true" autofit="true" bold="true">Header1<
/th>
     <th fontHeight="12" forceString="true" autofit="true" bold="true">Header2<
/th>
   </tr>
   <tr>
     <td>Value1</td>
     <td>Value11</td>
   </tr>
   <tr>
     <td>Value2</td>
     <td>Value22</td>
   </tr>
   <tr>
     <td>Value3</td>
     <td>Value33</td>
   </tr>
 </table>
</root>
```

Mit dem Attribut `forceString="true"` wird der Inhalt nicht als Zahl oder Datum interpretiert, sondern direkt in das Excel-Dokument reingeschrieben.
Beispiel: *`008.07300`* wird ohne das Attribut `forceString="true"` als Zahl *`8.073`* in das Excel-Dokument geschrieben, mit dem Attribut `forceString="true"` wird es als Zeichenkette *`008.07300`* in das Excel-Dokument geschrieben.

- **Operation `Fill Template`:**
  Für diese Operation erwartet der Adapter folgende XML-Struktur. In optionalen Attribut `templateUrl=".."` können Sie hierbei eine gültige URL zu einer Vorlagendatei angeben, z. B. *`templateUrl="file:///C:/Template.xlsx"`*. Diese Angabe überschreibt den Wert des Adapter-Parameters `TemplateURL`. Jedes Element `<cell>` entspricht einer Zelle der Tabelle. Für Zeilen- und ggf. Spaltennummern beginnt die Zählung bei 0.

Die Formatierung der einzelnen Zellen lässt sich über verschiedene Attribute steuern. Diese werden im Abschnitt **Mögliche Attribute** erläutert.

Beispiel-Input für Operation FillTemplate

```
<TemplateFill templateUrl="URL to template file">
    <cell  sheet = "Name of the spreadsheet"
            row = "Row number"
            col = "Column ID "
            value = "Cell content"
            formatNumber = "Format number"
            wrapText="true"/>
</TemplateFill>
```

Mögliche Attribute

Die Formatierung der einzelnen Zellen lässt sich über folgende Attribute steuern:

Attribut	Beschreibung
formatNumber	Zahlenformat **Mögliche Werte:** • `number1`: Zahl 1 • `number2`: Zahl 2 • `number3`: Zahl 3 • `number4`: Zahl 4 • `currency1`: Währung 1 • `currency2`: Währung 2 • `currency3`: Währung 3 • `currency4`: Währung 4 • `accounting`: Buchhaltung • `date`: Datum • `time`: Uhrzeit • `percentage`: Prozent • `fraction`: Bruch • `scientific`: Wissenschaft • `text`: Text • `postalCode`: Postleitzahl • `telephone`: Telefonnummer Bei Bedarf lassen sich auch benutzerdefinierte Zahlenformate hinterlegen
fontName	Schriftart, z.B. `Calibri`

horizontalAlignment	Horizontale Ausrichtung des Textes innerhalb der Zelle **Mögliche Werte:** • `CENTER` • `CENTER_SELECTION` • `DISTRIBUTED` • `FILL` • `GENERAL` • `JUSTIFY` • `LEFT` • `RIGHT`
verticalAlignment	Vertikale Ausrichtung des Textes innerhalb der Zelle **Mögliche Werte:** • `CENTER` • `BOTTOM` • `TOP` • `DISTRIBUTED` • `JUSTIFY`
fontHeight	Schriftgrad
fontColor	Schriftfarbe **Mögliche Werte:** • CSS-Farbangaben, z.B. `Red`, `Blue`, `Green` etc. • RGB-Farbcode, z.B. `240, 80, 0` • HEX-Farbcode, z.B. `#124361`
bold	Fette Schrift **Mögliche Werte:** • `true`: fette Schrift • `false`: normale Schrift
italic	Kursive Schrift **Mögliche Werte:** • `true`: kursive Schrift • `false`: normale Schrift
underline	Art des Unterstrichs **Mögliche Werte:** • `NONE` • `DOUBLE` • `DOUBLE_ACCOUNTING` • `SINGLE` • `SINGLE_ACCOUNTING`

cellColor	Füllfarbe der Zelle **Mögliche Werte:** • CSS-Farbangaben, z.B. `Red`, `Blue`, `Green` etc. • RGB-Farbcode, z.B. `240, 80, 0` • HEX-Farbcode, z.B. `#124361`
border **borderLeft** **borderRight** **borderBottom**	Art des Zellrahmens **Mögliche Werte:** • `DASH_DOT` • `DASH_DOT_DOT` • `DASHED` • `DOTTED` • `DOUBLE` • `HAIR` • `MEDIUM` • `MEDIUM_DASH_DOT` • `MEDIUM_DOT_DOT` • `MEDIUM_DASHED` • `SLANTED_DASH_DOT` • `THICK` • `THIN` • `NONE`
borderColor **BorderColorLeft** **borderColorRight** **borderColorBottom** **borderColorTop**	Farbe des Zellrahmens **Mögliche Werte:** • Farbname, z.B. `Red`, `Blue`, `Green` etc. • RGB-Farbcode, z.B. `240, 80, 0` • HEX-Farbcode, z.B. `#124361`
orientation	Text-Ausrichtung **Mögliche Werte:** • Zahl zwischen *`90`* und *`-90`* • `HORIZONTAL` • `VERTICAL`

wrapText	*Optionales Attribut*: Automatischen Zeilenumbruch in Zellen erlauben **Mögliche Werte:** • `true`: Zeilenumbruch aktivieren • `false`: Zeilenumbruch deaktivieren
autofit	*Optionales Attribut*: Spaltenbreite automatisch anpassen **Mögliche Werte:** • `true`: Automatische Spaltenbreite aktivieren • `false`: Automatische Spaltenbreite deaktivieren

Setzen des Attributs

Bitte beachten Sie:

- Für die Operation `Create Spreadsheet` muss das Attribut innerhalb der ersten Zeile im Element `<td>` oder `<th>` gesetzt werden.
- Für die Operation `Fill Template` muss zunächst ein neues Element `columnProperties` eingefügt werden, in dem die Konfiguration der einzelnen Spalten – jeweils in einem Knoten `column` – vorgenommen wird.

`columnProperties`	Gruppiert die Spalten eines Blattes. **Mögliche Werte:** `sheet`: Name des Excel-Blattes für die die Konfiguration vorgenommen wird.
`column`	Spalte innerhalb eines Blattes **Mögliche Werte:** `id`: Index der Spalte. Der erste Index ist `0`. Wird ein negativer Wert angegeben, wird eine Fehlermeldung ausgegeben.

```
<columnProperties sheet="Week">
    <column id="1" autofit="true" width="256" />
    <column id="11" width="500" />
    <column id="9" width="255" />
    <column id="7" width="22" />
    <column id="22" autofit="true" />
</columnProperties>
```

width *Optionales Attribut*: Feste Spaltenbreite definieren

> Ist das Attribut `autofit` gesetzt, wird das Attribut `width` ignoriert.

Mögliche Werte: Zahl zwischen *`0`* und *`255`*. Ist die angegebene Zahl größer `255`, wird `255` als maximale Spaltenbreite gesetzt.

Setzen des Attributs

Bitte beachten Sie:

- Für die Operation `Create Spreadsheet` muss das Attribut innerhalb der ersten Zeile im Element `<td>` oder `<th>` gesetzt werden.
- Für die Operation `Fill Template` muss zunächst ein neues Element `columnProperties` eingefügt werden, in dem die Konfiguration der einzelnen Spalten – jeweils in einem Knoten `column` – vorgenommen wird.

`columnProperties`	Gruppiert die Spalten eines Blattes. **Mögliche Werte:** `sheet`: Name des Excel-Blattes für die die Konfiguration vorgenommen wird.
`column`	Spalte innerhalb eines Blattes **Mögliche Werte:** `id`: Index der Spalte. Der erste Index ist `0`. Wird ein negativer Wert angegeben, wird eine Fehlermeldung ausgegeben.

```
<columnProperties sheet="Week">
    <column id="1" autofit="true" width="256" />
    <column id="11" width="500" />
    <column id="9" width="255" />
    <column id="7" width="22" />
    <column id="22" autofit="true" />
</columnProperties>
```

height	*Optionales Attribut*: Zeilenhöhe definieren

> Excel wendet die automatische Anpassung der Höhe mit dem Attribut `autofit` nur an, wenn zuvor eine feste Zeilenhöhe über das Attribut `height` definiert wurde. Ist das Attribut `autofit` für die Zeile gesetzt, wird das Attribut `heigth` ignoriert. Zusätzlich wird die automatische Zeilenhöhe auch über das Attribut `wrapText` geregelt.

Mögliche Werte: Zahl zwischen *`0`* und *`409`*. Ist die angegebene Zahl größer `409`, wird `409` als maximale Zeilenhöhe gesetzt.

> **Setzen des Attributs**
>
> **Bitte beachten Sie:**
>
> - Für die Operation `Create Spreadsheet` muss das Attribut innerhalb des Element `<tr>` gesetzt werden.
> - Für die Operation `Fill Template` muss zunächst ein neues Element `rowProperties` eingefügt werden, in dem die Konfiguration der einzelnen Zeilen – jeweils in einem Knoten `row` – vorgenommen wird.

`rowProperties`	Gruppiert die Zeilen eines Blattes. **Mögliche Werte:** `sheet`: Name des Excel-Blattes für die die Konfiguration vorgenommen wird.
`row`	Zeile innerhalb eines Blattes **Mögliche Werte:** `id`: Index der Zeile. Der erste Index ist `0`. Wird ein negativer Wert angegeben, wird eine Fehlermeldung ausgegeben.

```
<rowProperties sheet="Week">
   <row id ="7" height="50"/>
   <row id="6" height="409"/>
   <row id="8" autofit="true" height="250"/>
</rowProperties>
```

10.15.2 Microsoft Word Properties

? Unknown Attachment `Microsoft Word Properties`: Liest Metadaten (Dokumenteigenschaften wie Ersteller, letzte Speicherung) eines Microsoft Word-Dokuments (`.doc` und `.docx`) und gibt diese in einem UTF-8-kodierten XML-Dokument aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *Get*: Dokumenteigenschaften aus einem MS-Word-Dokument lesen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.wordheaderadapter.wordHeaderAdapter*: Hauptklasse (Standard)

Statuswerte

1 (successful)	Der Funktions-Adapter wurde erfolgreich ausgeführt und die ausgelesenen Dokumenteigenschaften wurden als XML-Dokument ausgegeben
0	Das verarbeitete Dokument hat ein falsches Dateiformat
-1 (error)	Das verarbeitete Dokument ist keine gültige `.doc` bzw `.docx`-Datei oder ein anderer Fehler ist aufgetreten (siehe Server-Protokoll).

Input

Der Adapter erwartet als Input ein Microsoft Word-Dokument mit der Endung `.doc` oder `.docx`.

Ausgabe

Der Adapter gibt ein XML-Dokument mit folgender Struktur aus:

```
<WordHeader>
    <Properties>
        <Author>Max Mustermann</Author>
        <Template>Normal.dotm</Template>
        <LastSavedBy>Max Mustermann</LastSavedBy>
        <LastPrinted>2017-12-07T09:24:00Z</LastPrinted>
        <ModifiedOn>2017-12-07T08:24:00Z</ModifiedOn>
        <Revision>3</Revision>
        <EditTime>60</EditTime> <!-- one hour editied in word -->
        <CreatedOn>2017-12-07T08:22:00Z</CreatedOn>
        <Title>Titelproperty</Titel>
        <Subject>Subjectproperty</Subject>
        <Keywords>Keyword</Keywords>
        <Comments>Kommentar</Comments>
        <Company>Max Mustermann GmbH</Company>
    </Properties>
    <Statistics>
        <Pages>1</Pages>
        <Words>6</Words>
        <Characters>40</Characters>
        <Application>Microsoft Office Word</Application>
        <Lines>1</Lines>
        <Paragraphs>1</Paragraphs>
        <CharactersWithSpaces>45</CharactersWithSpaces>
    </Statistics>
    <UserProperties />
</WordHeader>
```

Erläuterung: Das Element `UserProperties` ist nur bei Dateien mit der Endung `.doc` mit Daten befüllt.

> **Änderungen am Datumsformat**
> Bitte beachten Sie, dass sich bei der Ausgabe das Datumsformat geändert hat. Das zuvor deutsche Format `dd.mm.yyyy hh:mm:ss` wird nun nach ISO-8601 in der Zeitzone UTC formatiert `yyyy-mm-ddThh:mm:ssZ`.

10.15.3 Microsoft Word to Text Converter

? Unknown Attachment `Microsoft Word to Text Converter`: Liest ein *Microsoft Word* -Dokument (`.doc` im Format *Office 97-2003*) ein, extrahiert dessen Textinhalte und gibt ein Textdokument aus. Bei der Konvertierung werden alle Word-spezifischen Formatierungsparameter entfernt, so dass der reine Inhalt zur Weiterverarbeitung zur Verfügung steht.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *Convert*: Text aus einem MS-Word-Dokument extrahieren

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.Word2XML*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.15.4 Working Time Calculator

`Working Time Calculator`: Berechnet die Arbeitszeit.

> Der *Working Time Calculator* ersetzt mit Version 6.0 der X4 BPMS die veralteten Arbeitszeit-Berechnungsfunktionen in der Java-Klasse `de.softproject.xsl.WorkingTimeCalculator`, welche zur Verwendung innerhalb von XSL-Mappings bereitstellt wurden.

Eigenschaften (Properties)

`Operationen`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `AddWorkingTimeMinutes`: Addiert zu einem Start-Datum eine bestimmte Anzahl an Arbeitsminuten und gibt das End-Datum und die Uhrzeit aus • `GetWorkingTimeMinutes`: Berechnet die Arbeitszeit zwischen einem Start-Datum und einem End-Datum und gibt die Arbeitsminuten aus • `GetEaster`: Berechnet den Ostersonntag anhand einer Jahreszahl

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.adapter.workingTimeCalculator*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt und in der Ausgabe befindet sich ein XML-Dokument mit dem Resultat zur jeweiligen Operation

`-1`	Es kam zu einem Fehler während der Ausführung: • Das erforderliche Datums-Format (`yyyy-MM-dd` bzw. `yyyy-MM-dd hh:mm`) wurde nicht eingehalten • Die Wochentage wurden nicht richtig eingetragen. Erforderlich ist ein korrekter deutscher oder englischer Wochentag. • Es ist kein Input vorhanden

Input

Der Adapter erwartet eine vordefinierte Input-XML-Struktur für die verschiedenen Operationen

- Feiertage bzw. Urlaubstage (Arbeitsfreie Tage) werden im Datumsformat `yyyy-MM-dd` innerhalb beliebig vieler `<Holiday>`-Elemente definiert. Diese werden als Kind-Elemente von `<Holidays>` angelegt.
- Arbeitstage (Wochentage) werden innerhalb von `<WorkingDay>`-Elementen definiert. Diese werden als Kind-Elemente von `<WorkingDays>` angelegt. Der Wert von `<WorkingDay>` ist der jeweilige Wochentag auf deutsch oder auf englisch, z.B. *`Montag`* oder *`Friday`*. Dabei ist die Groß- und Kleinschreibung egal.
- Spezielle Tage werden im Datumsformat `yyyy-MM-dd hh:mm-hh:mm` innerhalb beliebig vieler `<SpecialDay>`-Elemente definiert. Diese werden als Kind-Elemente von `<SpecialDays>` angelegt.

  > `SpecialDays` gelten als spezieller Arbeitstag unabhängig davon, ob es sich um einen normalen Arbeitstag oder arbeitsfreien Tag handelt. Als Arbeitszeit zählt nur die Zeit, die innerhalb von `<SpecialDay>` definiert wurde.

> Die Arbeitszeit-Berechnungsfunktionen `dateDiff` oder `dateDiffWorkingMinutes` der veralteten Java-Klasse `de.softproject.xsl.WorkingTimeCalculator` wurden durch folgende Funktionen ersetzt:
>
> - `dateDiffWorkingMinutes` kann durch die Operation `GetWorkingTimeMinutes` umgesetzt werden
> - `dateDiff` lässt sich umsetzten, indem alle Wochentage als `WorkingDay` definiert und die Arbeitszeiten von `00:00` bis `23:59` einstellt werden. Dabei dürfen weder `Holidays` noch `SpecialDays` definiert werden.

- Operation `AddWorkingTimeMinutes`:

Beispiel-Input für die Operation AddWorkingTimeMinutes

```
<WorkingTimeCalculator>
    <Holidays>
        <Holiday>2019-01-01</Holiday>
        <Holiday>2019-01-06</Holiday>
        <Holiday>2019-04-19</Holiday>
        <Holiday>2019-04-21</Holiday>
        <Holiday>2019-04-22</Holiday>
    </Holidays>
    <SpecialDays>
        <SpecialDay>2019-12-24 21:00-23:00</SpecialDay>
    </SpecialDays>
    <!--You can use either the German or the English word-->
    <WorkingDays>
        <WorkingDay>Monday</WorkingDay>
        <WorkingDay>Tuesday</WorkingDay>
        <WorkingDay>Wednesday</WorkingDay>
        <WorkingDay>Thursday</WorkingDay>
        <WorkingDay>Friday</WorkingDay>
    </WorkingDays>
    <setWorkingDayBegin>09:30</setWorkingDayBegin>
    <setWorkingDayEnd>17:30</setWorkingDayEnd>
    <StartDate>2019-12-23 10:00</StartDate> <!-- Format yyyy-MM-dd or yyyy-MM-
dd hh:mm -->
    <addMinutesToWorkingTime>500</addMinutesToWorkingTime>
</WorkingTimeCalculator>
```

- Operation `GetWorkingTimeMinutes`:

Beispiel-Input für die Operation GetWorkingTimeMinutes

```
<WorkingTimeCalculator>
    <Holidays>
        <Holiday>2019-01-01</Holiday>
        <Holiday>2019-01-06</Holiday>
        <Holiday>2019-04-19</Holiday>
        <Holiday>2019-04-21</Holiday>
        <Holiday>2019-04-22</Holiday>
    </Holidays>
    <SpecialDays>
        <SpecialDay>2019-12-24 21:00-23:00</SpecialDay>
    </SpecialDays>
    <!--You can use either the German or the English word-->
    <WorkingDays>
        <WorkingDay>Monday</WorkingDay>
        <WorkingDay>Tuesday</WorkingDay>
        <WorkingDay>Wednesday</WorkingDay>
        <WorkingDay>Thursday</WorkingDay>
        <WorkingDay>Friday</WorkingDay>
    </WorkingDays>
    <StartDate>2019-12-23 10:00</StartDate> <!-- Format yyyy-MM-dd or yyyy-MM-
dd hh:mm -->
    <EndDate>2020-01-01 10:00</EndDate> <!-- Format yyyy-MM-dd or yyyy-MM-dd
hh:mm -->
    <setWorkingDayBegin>09:30</setWorkingDayBegin>
    <setWorkingDayEnd>17:30</setWorkingDayEnd>
</WorkingTimeCalculator>
```

- Operation `GetEaster`:

Beispiel-Input für die Operation GetEaster

```
<WorkingTimeCalculator>
    <GetEaster>2019</GetEaster>
</WorkingTimeCalculator>
```

Ausgabe

Der Adapter gibt je nach Operation ein XML-Dokument aus.

- Operation `AddWorkingTimeMinutes`:

Beispiel-Output für die Operation AddWorkingTimeMinutes

```
<WorkingTimeCalculator>
    <addWorkingTimeMinutes>2019-12-24T21:50:00</addWorkingTimeMinutes>
</WorkingTimeCalculator>
```

- Operation `GetWorkingTimeMinutes`:

Beispiel-Output für die Operation GetWorkingTimeMinutes

```
<WorkingTimeCalculator>
    <getWorkingTimeMinutes>2010.0</getWorkingTimeMinutes>
</WorkingTimeCalculator>
```

- Operation `GetEaster`:

Beispiel-Output für die Operation GetEaster

```
<EasterHolidays>
    <EasterSunday>2019-04-21</EasterSunday>
</EasterHolidays>
```

10.16 PDF Processing

- PDF Data Processing
- PDF Image Processing
- PDF Split and Merge

10.16.1 PDF Data Processing

10.16.1.1 PDF Barcode Adder

? Unknown Attachment `PDF Barcode Adder`: Erzeugt Strichcodes und fügt diese einem bestehenden nicht geschützten PDF-Dokument hinzu. Die Strichcode-Grafiken können auf jeder PDF-Seite positioniert, skaliert und gedreht werden. Unterstützt werden die Strichcode-Typen *`Code 39`*, *`Code 128`*, *`Codabar`*, *`EAN 13`*, *`Code 25-Interleaved 2 of 5`* sowie *`POSTNET`*.
In einem adapterspezifisch aufgebauten Input-XML-Dokument werden die Daten der Strichcodes hinterlegt. Der Adapter gibt ein PDF-Dokument aus, das um die angegebenen Strichcodes ergänzt wurde.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`AddBarcodes`*: Dem PDF-Dokument Strichcode(s) hinzufügen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.pdf.PDFBarcode*: Hauptklasse (Standard)
ReadPartially	PDF-Dokument in mehreren Teilen in den Arbeitsspeicher laden (kann Performance-Verbesserungen bringen) **Mögliche Werte:** • *ja*: In mehreren Teilen nacheinander in den Arbeitsspeicher laden • *nein*: PDF-Dokument In einem Stück in den Arbeitsspeicher laden (Standard)
ToXML	PDF-Dokument Base64-kodiert in einem XML-Dokument ausgeben **Mögliche Werte:** • *ja*: XML-Dokument ausgeben • *nein*: PDF-Dokument als Binärdatenstrom ausgeben (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Funktions-Adapter erwartet eine adapterspezifische Input-XML-Struktur, in der sämtliche Daten jeder Strichcode-Grafik hinterlegt sind (siehe unten). Wenn ein PDF aus dem Repository als Quelldokument verwendet werden soll, in Element `<PDF>` im Attribut `url` dessen `xstore`-URL angeben. Alternativ, wenn das Quelldokument als Base64-kodierte Zeichenkette vorliegt, können Sie diese Zeichenkette als Inhalt des Elements `<PDF encoding="base64">` hinterlegen (siehe auskommentierte Zeile).

```
<pdfBarcode>
   <PDF url="xstore-URL des Input-PDF-Dokuments im Repository"/>
   <!-- <PDF encoding="base64">(Wenn Input-PDF Base64-Zeichenkette)</PDF> -->
   <Barcode height="Höhe" width="Breite"
   rad="Drehung (Bogenmaß)" degree="Drehung (Grad)"
   xPos="Abstand vom linken Seitenrand" yPos="Abstand vom unteren Seitenrand"
   type="Strichcode-Typ" page="Seite">Strichcode-Daten</Barcode>
</pdfBarcode>
```

Attribute im Input-XML-Dokument

Der *PDF Barcode Adder* erwartet eine adapterspezifische Input-XML-Struktur nach oben genanntem Muster. Folgende Attribute können gesetzt werden:

url	xstore-Pfadangabe zum Input-PDF-Dokument im Repository (z. B. *xstore://Project/Data/InputData.xml*; erforderlich)
height	Höhe der Strichcode-Grafik in *cm* oder *pt* (erforderlich)
width	Breite der Strichcode-Grafik in *cm* oder *pt* (erforderlich)
rad	Drehung der Strichcode-Grafik im Bogenmaß
degree	Drehung der Strichcode-Grafik in Grad (erforderlich)
xPos	Abstand der Strichcode-Grafik vom linken Seitenrand in *cm* oder *pt* (erforderlich)
yPos	Abstand der Strichcode-Grafik vom unteren Seitenrand in *cm* oder *pt* (erforderlich)
type	Strichcode-Typ (erforderlich) Mögliche Werte: • *39*: *Code 39* -Format • *128*: *Code 128* -Format • *Codabar*: *Codabar* -Format • *EAN*: *EAN 13* -Format • *Inter25*: *Code 25 – Interleaved 2 of 5* -Format • *Postnet*: *POSTNET* -Format
page	Seitenzahl im PDF-Dokument, auf der die Strichcode-Grafik positioniert wird (erforderlich)

Beispiel

Folgendes Input-XML-Dokument enthält die Daten zum Erzeugen einer *EAN 13*-Strichcode-Grafik mit 3 cm Breite und 20° Neigung auf der ersten Seite eines PDF-Dokuments im Repository.

```
<pdfBarcode>
   <PDF url="xstore://Project/Folder/InputPDF.pdf"/>
   <!-- <PDF encoding="base64">base_64_encoded_PDF_document</PDF> -->
   <Barcode height="3cm" width="3cm" rad="" degree="20"
         xPos="120pt" yPos="1cm" type="EAN"
         page="1">1234567891234</Barcode>
</pdfBarcode>
```

10.16.1.2 PDF Form

`PDF Form`: Erstellt aus FDF/XFDF-Formulardaten ein PDF-Dokument mit oder ohne Formularfelder, extrahiert eine FDF-Datei aus einem PDF-Formulardokument, setzt Formularfeldwerte oder wandelt ein PDF-Formulardokument in ein flaches PDF-Dokument um. Ehemalige Bezeichnung bis Version 4.5.1: *PDF Form Processing Adapter*

> Die FDF/XFDF-Formulardaten werden in einem Text- bzw. XML-Dokument gespeichert. Dieses Formulardatendokument muss jeweils eine absolute Pfadangabe zu einem PDF-Formular-Dokument enthalten, das als Vorlage für das zu erstellende PDF-Dokument dient (nicht relevant für die Operationen `Export FDF` und `Flatten`).

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • `Flatten XFDF`: XFDF-Daten in ein PDF-Formulardokument schreiben (Formularfelder in Text umwandeln) • `Flatten FDF`: FDF-Daten in ein PDF-Formulardokument schreiben (Formularfelder in Text umwandeln) • `Export FDF`: FDF-Daten aus einem PDF-Formulardokument extrahieren • `Import XFDF`: XFDF-Daten in ein PDF-Formulardokument schreiben (Formularfelder erhalten) • `Import FDF`: FDF-Daten in ein PDF-Formulardokument schreiben (Formularfelder erhalten) • `Set Fields`: Formularfelderwerte setzen und ggf. im ausgegebenen PDF-Dokument die Formularfelder in Text umwandeln • `Flatten`: PDF-Formulardokument in ein flaches PDF-Dokument umwandeln (Formularfelder in Text umwandeln)

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.pdf.PDFFormular`*: Hauptklasse (Standard)
`PreserveExtendedAttributes`	Beim Verwenden der Operation `Set Fields` die erweiterten Attribute im PDF-Dokument erhalten, wenn für ein Formularfeld im Input-XML über `flatten="false"` definiert wurde, dass Formularfelder editierbar bleiben sollen **Mögliche Werte:** • *`ja`*: Nach dem Befüllen der Formularfelder die erweiterten Attribute erhalten (Standard) • *`nein`*: Erweiterte Attribute für die Formularfelder im PDF-Dokument nicht erhalten (dies entspricht dem Verhalten in vorherigen Versionen dieses Adapters und ist nur bei Kompatibilitätsproblemen relevant)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.16.1.2.1 Operation: Flatten XFDF

Die Operation `Flatten XFDF` schreibt XFDF-Daten (ISO 19444-1) in ein PDF-Formulardokument. Die Formularfelder werden in Text umgewandelt und können nicht editiert werden.

Input

> Der Aufbau eines XFDF-Dokuments ist in der ISO 19444-1 beschrieben.

Für Operation `Flatten XFDF` erwartet der Adapter ein XFDF-Dokument, das den absoluten Pfad zum zugehörigen PDF-Formular in Element `<f>` angibt.

```
<?xml version="1.0" encoding="UTF-8"?>
<xfdf xmlns:math="java.lang.Math" xmlns:cal="java.util.Calendar"
    xmlns:sdf="java.text.SimpleDateFormat" xmlns="http://ns.adobe.com/xfdf/">
    <fields>
        <field name="Firma">
            <value>SoftProject GmbH</value>
        </field>
        <field name="KDNr">
            <value>123567234</value>
        </field>
        <field name="Ort">
            <value>Ettlingen</value>
        </field>
        <field name="PLZ">
            <value>76275</value>
        </field>
        <field name="Strasse">
            <value>Am Erlengraben 3</value>
        </field>
    </fields>
    <f
    href="X4DB/1/Project/Resources/FormDocument.pdf" />
</xfdf>
```

Output

Als Output wird eine Kopie des im Input referenzierten PDF-Formulardokuments mit flachem Text erzeugt.

Weitere Informationen finden Sie im Abschnitt

- Operation: Flatten XFDF
- Operation: Flatten FDF
- Operation: Export FDF
- Operation: Import XFDF
- Operation: Import FDF
- Operation: Set Fields
- Operation: Flatten

10.16.1.2.2 Operation: Flatten FDF

Die Operation `Flatten FDF` schreibt FDF-Daten (ISO 19444-1) in ein PDF-Formulardokument. Die Formularfelder werden in Text umgewandelt und können nicht editiert werden.

Input

Für Operation `Flatten FDF` erwartet der Adapter ein FDF-Dokument, das den absoluten Pfad zum zugehörigen PDF-Formular im Parameter `/F` angibt.

```
%FDF-1.2
1 0 obj
<</FDF<</Fields[<</T(Firma)/V(SoftProject GmbH)>><</T(Ort)/V(Ettlingen)>><</T(Strasse
)/V(Am Erlengraben 3)>><</T(KDNr)/V(123567234)>><</T(PLZ)/V(76275)>>]/F(X4DB/1/
Project/Resources/FormDocument.pdf)>>>>
endobj
trailer
<</Root 1 0 R>>
%%EOF
```

Output

Als Output wird eine Kopie des im Input referenzierten PDF-Formulardokuments mit flachem Text erzeugt.

Weitere Informationen finden Sie im Abschnitt

- Operation: Flatten XFDF
- Operation: Flatten FDF
- Operation: Export FDF
- Operation: Import XFDF
- Operation: Import FDF
- Operation: Set Fields
- Operation: Flatten

10.16.1.2.3 Operation: Export FDF

Die Operation `Export FDF` extrahiert FDF-Daten (ISO 19444-1) aus einem PDF-Formulardokument.

Input

Für Operation `Export FDF` erwartet der Adapter ein PDF-Formulardokument.

Output

Die im PDF-Formulardokument enthaltenen FDF-Daten werden ausgegeben.

Weitere Informationen finden Sie im Abschnitt

- Operation: Flatten XFDF
- Operation: Flatten FDF
- Operation: Export FDF
- Operation: Import XFDF
- Operation: Import FDF
- Operation: Set Fields
- Operation: Flatten

10.16.1.2.4 Operation: Import XFDF

Die Operation `Import XFDF` schreibt XFDF-Daten (ISO 19444-1) in ein PDF-Formulardokument. Die Formularfelder bleiben erhalten und können weiterhin editiert werden.

Input

Der Aufbau eines XFDF-Dokuments ist in der ISO 19444-1 beschrieben.

Für Operation `Import XFDF` erwartet der Adapter ein XFDF-Dokument, das den absoluten Pfad zum zugehörigen PDF-Formular in Element `<f>` angibt.

```
<?xml version="1.0" encoding="UTF-8"?>
<xfdf xmlns:math="java.lang.Math" xmlns:cal="java.util.Calendar"
    xmlns:sdf="java.text.SimpleDateFormat" xmlns="http://ns.adobe.com/xfdf/">
    <fields>
        <field name="Firma">
            <value>SoftProject GmbH</value>
        </field>
        <field name="KDNr">
            <value>123567234</value>
        </field>
        <field name="Ort">
            <value>Ettlingen</value>
        </field>
        <field name="PLZ">
            <value>76275</value>
        </field>
        <field name="Strasse">
            <value>Am Erlengraben 3</value>
        </field>
    </fields>
    <f
    href="X4DB/1/Project/Resources/FormDocument.pdf" />
</xfdf>
```

Output

Als Output wird eine Kopie des im Input referenzierten PDF-Formulardokuments mit ausgefüllten Formularfeldern erzeugt.

Weitere Informationen finden Sie im Abschnitt

- Operation: Flatten XFDF
- Operation: Flatten FDF
- Operation: Export FDF
- Operation: Import XFDF
- Operation: Import FDF
- Operation: Set Fields
- Operation: Flatten

10.16.1.2.5 Operation: Import FDF

Die Operation `Import FDF` schreibt FDF-Daten (ISO 19444-1) in ein PDF-Formulardokument. Die Formularfelder bleiben erhalten und können weiterhin editiert werden.

Input

Für Operation `Import FDF` erwartet der Adapter ein FDF-Dokument, das den absoluten Pfad zum zugehörigen PDF-Formular im Parameter `/F` angibt.

```
%FDF-1.2
1 0 obj
<</FDF<</Fields[<</T(Firma)/V(SoftProject GmbH)>>><</T(Ort)/V(Ettlingen)>>><</T(Strasse
)/V(Am Erlengraben 3)>>><</T(KDNr)/V(123567234)>>><</T(PLZ)/V(76275)>>]/F(X4DB/1/
Project/Resources/FormDocument.pdf)>>>>
endobj
trailer
<</Root 1 0 R>>
%%EOF
```

Output

Als Output wird eine Kopie des im Input referenzierten PDF-Formulardokuments mit ausgefüllten Formularfeldern erzeugt.

Weitere Informationen finden Sie im Abschnitt

- Operation: Flatten XFDF
- Operation: Flatten FDF
- Operation: Export FDF
- Operation: Import XFDF
- Operation: Import FDF
- Operation: Set Fields
- Operation: Flatten

10.16.1.2.6 Operation: Set Fields

Die Operation `Set Fields` setzt Formularfeldwerte im ausgegebenen PDF-Dokument. Formularfelder werden im ausgegebenen PDF-Dokument in Text umgewandelt und sind nicht mehr editierbar, wenn das Attribut `flatten="true"` deklariert wurde.

Input

Für Operation `Set Fields` erwartet der Adapter ein adapterspezifisches XML-Dokument, das den absoluten Pfad zum zugehörigen PDF-Formular in Element `<Document>` und die zu setzenden Formularfelddaten in beliebig vielen Elementen `<Field>` angibt. Mit Attribut `flatten="true"` werden im erzeugten PDF-Dokument die Formularfelder in flachen Text umgewandelt, mit `flatten="false"` bleiben die Formularfelder editierbar.

```
<?xml version="1.0" encoding="UTF-8"?>
<Input>
   <Document flatten="true">X4DB/1/Proect/Resources/FormDocument.pdf</Document>
   <Field name="Firma">SoftProject GmbH</Field>
   <Field name="KDNr">123567234</Field>
   <Field name="Ort">Ettlingen</Field>
   <Field name="PLZ">76275</Field>
   <Field name="Strasse">Am Erlengraben 3</Field>
</Input>
```

Output

Als Output wird eine Kopie des im Input referenzierten PDF-Formulardokuments mit flachem Text erzeugt. Der Text kann nicht editiert werden, da im Input `flatten="true"` definiert wurde.

Weitere Informationen finden Sie im Abschnitt

- Operation: Flatten XFDF
- Operation: Flatten FDF
- Operation: Export FDF
- Operation: Import XFDF
- Operation: Import FDF
- Operation: Set Fields
- Operation: Flatten

10.16.1.2.7 Operation: Flatten

Die Operation *`Flatten`* wandelt ein PDF-Formulardokument in ein PDF-Dokument um. Die Formularfelder werden in Text umgewandelt und können nicht editiert werden.

Input

Für Operation `Flatten` erwartet der Adapter ein PDF-Formulardokument.

Output

Als Output wird ein PDF-Dokument erzeugt.

Weitere Informationen finden Sie im Abschnitt

- Operation: Flatten XFDF
- Operation: Flatten FDF
- Operation: Export FDF
- Operation: Import XFDF
- Operation: Import FDF
- Operation: Set Fields
- Operation: Flatten

10.16.1.3 PDF Metadata

`PDF Metadata`: Liest Metadatenstrukturen aus PDF-Dokumenten aus oder schreibt Metadatenstrukturen in PDF-Dokumente.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • `GetMetadata`: Metadaten auslesen • `SetMetadata`: Metadaten hinzufügen/ersetzen

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** `de.softproject.integration.adapter.pdf.PDFMetadata` Hauptklasse (Standard)
`ReadPartially`	PDF-Dokument in mehreren Teilen in den Arbeitsspeicher laden (kann Performance-Verbesserungen bringen) **Mögliche Werte:** • *`ja`*: In mehreren Teilen nacheinander in den Arbeitsspeicher laden • *`nein`*: PDF-Dokument(e) in einem Stück in den Arbeitsspeicher laden (Standard)
`ToXML`	PDF-Dokument Base64-kodiert in einem XML-Dokument ausgeben **Mögliche Werte:** • *`ja`*: XML-Dokument ausgeben • *`nein`*: PDF-Dokument als Binärdatenstrom ausgeben (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Weitere Informationen finden Sie im Abschnitt

- Operation: GetMetadata
- Operation: SetMetadata

10.16.1.3.1 Operation: GetMetadata

Die Operation `GetMetadata` liest die Metadaten einer PDF-Datei aus.

Input

PDF im PDF-Format

Folgendes Input-XML-Dokument enthält den Pfad zu einem PDF-Dokument im Repository.

```
<?xml version="1.0" encoding="UTF-8"?>
<PDF url="xstore://Project/Folder/Input.pdf"/>
```

PDF im Base64-Format

Folgendes Input-XML-Dokument enthält die Base64-Zeichenkette.

```
<?xml version="1.0" encoding="UTF-8"?>
<PDF encoding="base64">
    Base64-Zeichenkette
</PDF>
```

Output

Wenn der Parameter `ToXML` **deaktiviert** ist, sieht der Output wie folgt aus:

Output beim Parameter ToXML: ja

```
{CreationDate=D:20210610111922+02'00', ModDate=D:20210610111922+02'00'}
```

Wenn der Parameter `ToXML` **aktiviert** ist, sieht der Output wie folgt aus:

Output beim Parameter ToXML: ja

```
<?xml version="1.0" encoding="UTF-8"?>
<Metadata>
   <CreationDate>D:20210610111922+02'00'</CreationDate>
   <ModDate>D:20210610111922+02'00'</ModDate>
</Metadata>
```

Weitere Informationen finden Sie im Abschnitt

- Operation: GetMetadata
- Operation: SetMetadata

10.16.1.3.2 Operation: SetMetadata

Die Operation `SetMetadata` fügt Metadaten einer PDF-Datei hinzu oder ersetzt vorhandene Metadaten.

PDF Standard-Metadaten

Element	Beschreibung
`<Author>`	Autor des Dokuments
`<CreationDate>`	das Erstellungsdatum und die Uhrzeit
`<Creator>`	Die ursprüngliche Anwendung, die das Dokument erstellt hat

Element	Beschreibung
<Producer>	Das Produkt, das das Dokument erstellt hat.
<Subject>	Thema des Dokuments
<Title>	Titel des Dokuments
<Keywords>	Schlagwörter Können durch Kommas getrennt werden
<ModDate>	Datum und Uhrzeit der letzten Änderung

Benutzerdefinierte Metadaten fügen Sie mit einem benutzerdefiniertem Element hinzu. Im folgenden Beispiel wird ein Metadatum `Region` mit dem Wert `Europe` hinzugefügt:

```
<Region>Europe</Region>
```

Input

PDF im PDF-Format

Folgendes Input-XML-Dokument enthält den Pfad zu einem PDF-Dokument im Repository und die hinzuzufügenden Metadaten.

```
<?xml version="1.0" encoding="UTF-8"?>
<pdfMetadata>
   <PDF url="xstore://_MEINTEST/Resources/input.pdf" />
  <Metadata>
    <Author>SoftProject GmbH</Author>
    <CreationDate>D:20210610111922+02'00'</CreationDate>
    <Creator>X4 BPMS</Creator>
    <Producer>X4 BPMS</Producer>
    <Subject>X4 BPMS Output</Subject>
    <Title>Hello World</Title>
    <Keywords>keyword1;keyword2;keyword3</Keywords>
    <ModDate>D:20210610111922+02'00'</ModDate>
    <Region>Europe</Region>
  </Metadata>
</pdfMetadata>
```

PDF im Base64-Format

Folgendes Input-XML-Dokument enthält die Base64-Zeichenkette und die hinzuzufügenden Metadaten.

```
<?xml version="1.0" encoding="UTF-8"?>
<pdfMetadata>
  <PDF encoding="base64">
    Base64-Zeichenkette
  </PDF>
  <Metadata>
    <Author>SoftProject GmbH</Author>
    <CreationDate>D:20210610111922+02'00'</CreationDate>
    <Creator>X4 BPMS</Creator>
    <Producer>X4 BPMS</Producer>
    <Subject>X4 BPMS Output</Subject>
    <Title>Hello World</Title>
    <Keywords>keyword1;keyword2;keyword3</Keywords>
    <ModDate>D:20210610111922+02'00'</ModDate>
    <Region>Europe</Region>
  </Metadata>
</pdfMetadata>
```

Output

Wenn der Parameter `ToXML` **deaktiviert** ist, kann der Output direkt als Buffer-Datei im PDF-Format gespeichert werden.

Wenn der Parameter `ToXML` **aktiviert** ist, sieht der Output wie folgt aus:

Output beim Parameter ToXML: ja

```
<?xml version="1.0" encoding="UTF-8"?>
<PDF encoding="base64">
  Base64-Zeichenkette
</PDF>
```

Weitere Informationen finden Sie im Abschnitt

- Operation: GetMetadata
- Operation: SetMetadata

10.16.1.4 PDF to Text Converter

? Unknown Attachment `PDF to Text Converter`: Liest ein PDF-Dokument (auch passwortgeschützt) ein, extrahiert die Textinhalte eines bestimmten Seitenbereichs oder des gesamten Dokuments und gibt die extrahierten Zeichenketten in einem XML- oder Textdokument mit einer frei wählbaren Zeichenkodierung aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *`Extract`*: Text aus dem Input-PDF-Dokument extrahieren

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.pdf.PDF2Text`*: Hauptklasse (Standard)
password	Passwort (bei einem geschützten PDF-Dokument) **Mögliche Werte:** Beliebige Zeichenkette
startPage	Erste Seitenzahl, ab der die Texte extrahiert werden sollen **Mögliche Werte:** • Beliebige positive ganze Zahl oder *`0`* • *`0`*: Ab der ersten Seite beginnen (Standard)
endPage	Letzte Seitenzahl, bis zu der die Textextraktion durchgeführt werden soll **Mögliche Werte:** • Beliebige ganze Zahl oder *`0`* • *`0`*: Bis zur letzten Seite Text extrahieren (Standard)
encoding	Zeichenkodierung des Ergebnisdokuments **Mögliche Werte:** Beliebige gültige Zeichenkodierung (z.B. *`UTF-8`*)
force	Auch auf ungültigen PDF-Seiten versuchen, Text zu extrahieren **Mögliche Werte:** • *`ja`*: Ungültige PDF-Seiten verarbeiten • *`nein`*: Ungültige PDF-Seiten ignorieren (Standard)

toXML	Textinhalte in einem XML-Dokument ausgeben **Mögliche Werte:** • *ja*: XML-Dokument ausgeben • *nein*: Textdokument ausgeben (Standard)

Statuswerte

-1	Die Operation wurde erfolgreich ausgeführt.
1	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.16.2 PDF Image Processing

10.16.2.1 PDF Image Adder

`PDF Image Adder`: Fügt einem bestehenden nicht geschützten PDF-Dokument eine oder mehrere Grafiken hinzu oder erstellt ein neues PDF-Dokument mit den angegebenen Grafiken. Diese können sowohl als URL als auch als Base64-kodierte Zeichenkette angegeben werden.

Unterstützte Grafikformate sind BMP, GIF, JPEG/JPG, PNG und TIFF. Die Grafiken können auf jeder PDF-Seite positioniert, skaliert und gedreht werden. In einem adapterspezifisch aufgebauten Input-XML-Dokument werden die Informationen zu den Grafiken hinterlegt. Der Adapter gibt entweder ein neues PDF-Dokument oder ein PDF-Dokument, das um die angegebenen Grafiken ergänzt wurde, aus.

> Ab der Adapter-Version 2.0.0 werden die Grafikformate EPS und WMF nicht mehr unterstützt.

Enthält das Input-Dokument kein PDF-Element, wird zunächst ein PDF-Dokument im DIN A4-Format mit der definierten Anzahl an Seiten angelegt, bevor die Grafiken hinzugefügt werden.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *AddImages*: Dem PDF-Dokument Grafiken hinzufügen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.pdf.PDFImage*: Hauptklasse (Standard)

ReadPartially	PDF-Dokument in mehreren Teilen in den Arbeitsspeicher laden (kann Performance-Verbesserungen bringen) **Mögliche Werte:** • *ja*: In mehreren Teilen nacheinander in den Arbeitsspeicher laden • *nein*: PDF-Dokument in einem Stück in den Arbeitsspeicher laden (Standard)
ToXML	PDF-Dokument Base64-kodiert in einem XML-Dokument ausgeben **Mögliche Werte:** • *ja*: XML-Dokument ausgeben • *nein*: PDF-Dokument als Binärdatenstrom ausgeben (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Funktions-Adapter erwartet eine adapterspezifische Input-XML-Struktur, in der Informationen zu jeder Grafik hinterlegt sind. Hierzu zählen Breite und Höhe (in *cm* oder *pt*), Drehung, x- und y-Position (Abstände vom linken unteren Seitenrand in *cm* oder *pt*), der URL zur Grafik sowie die Seite, auf der die Grafik positioniert werden soll. Das Input-PDF-Dokument kann entweder als URL oder als Base64-kodierte Zeichenkette angegeben werden.

```
<pdfImage>
   <!-- <PDF url="xstore-URL des Input-PDF-Dokuments im Repository" (Wenn Input-PDF
als URL angegeben wird) /> -->
   <!-- <PDF encoding="base64"> Base64-Zeichenkette (Wenn Input-PDF Base64-
Zeichenkette)</PDF> -->
   <Image height="Höhe" width="Breite"
          rad="Drehung (Bogenmaß)" degree="Drehung (Grad)"
          xPos="Abstand vom linken Seitenrand" yPos="Abstand vom unteren Seitenrand"
          url="xstore-URL der Grafik im Repository" page="Seite im PDF"/>
</pdfImage>
```

Folgendes Input-XML-Dokument enthält Informationen für eine PNG-Bilddatei, die auf der ersten Seite eines bereits bestehenden PDF-Dokuments (Angabe als URL oder Base64-Zeichenkette) eingefügt werden soll.

```
<pdfImage>
   <!-- <PDF url="xstore://Project/Folder/InputPDF.pdf"/> -->
   <!-- <PDF encoding="base64">base_64_encoded_PDF_document</PDF> -->
   <Image height="90pt" width="60pt" rad="" degree=""
          xPos="400pt" yPos="100pt"
          url="xstore://Project/Folder/Image.png" page="1"/>
</pdfImage>
```

Folgendes Input-XML-Dokument enthält Informationen für drei Grafiken, die in einem neuen PDF-Dokument zusammengeführt werden sollen.

```
<?xml version="1.0" encoding="UTF-8"?>
<Root>
    <Image page="1" height="100pt" xPos="10pt" yPos="500pt" url="https://
www.softproject.de/assets/images/a/Logo-SoftProject_www-ee6b7fba.png"/>
    <Image page="3" height="100pt" xPos="10pt" yPos="500pt" url="https://
www.softproject.de/assets/images/a/Logo-SoftProject_www-ee6b7fba.png"/>
    <Image page="3" height="120pt" xPos="50pt" yPos="100pt" encoding="base64">

iVBORw0KGgoAAAANSUhEUgAAAEsAAAAWCAYAAACIXmHDAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAA
AAJcEhZcwAAEnQAABJ0Ad5mH3gAAAHDSURBVFhH7ZSxSgNBEIZ9gjyZjS/
hY9jbiJW9rWBlaWNpqVUqEQRBwSBRE4gxWf3X/
MfvuHvcbO66HfjZy+zO7Mx3k9t53huFqm6qsByqsByqsBz6B+v1YDesJo9hvVyEj7PDxj+/
PA0wrHp+erwf1vP3KDzrnu6nbPkwbvKq6d2L26uN99e4h9iU4a7Z+VHswRpiNCd/U/
AjDgzUTyUniw0ykABtcohF52CpUEzXl5DaQx1fT/d/4vGMnMhNH+vNNU5YtpYiWBATYoVSMNAE/
Mu7m0Fgpc5blcJaTV+idACivwQWL6TZZjh98MemBpysttzFsH72ZhcnMTfrKYYFsdhUArwR+vuApaa5EENL3d
EGyxrPYGXt6IPTpX7mUnWClWpQfbhkW1hY9azKNq9n22DlGlcoMX4zXcWw0DiS8HukSfhRT5l+A6xKYVEEo7V
sCwu/+S/5HF9nY6AsLCZAIJK3NRQhDDhZKtSld/UBq3kJbxM/
LAazAV6eAzIULJzXSW2mXXx9wIKQE+aClSoIYlPWD/
UByxr8OG9NoUBtsKwRBM5aKMzjglWVV4XlUIXlUIXlUIXlUIXlUIXVWaPwDd5omEQbL+NTAAAAAElFTkSuQmC
C</Image>
</Root>
```

10.16.2.2 PDF Multipage TIFF Adder

? Unknown Attachment `PDF Multipage TIFF Adder`: Fügt einem bestehenden nicht geschützten PDF-Dokument eine oder mehrere Grafiken aus mehrseitigen TIFF-Dateien (*Multipage Tagged Image File Format*) hinzu.

Die Grafiken können auf jeder PDF-Seite positioniert, skaliert und gedreht werden. In einem adapterspezifisch aufgebauten Input-XML-Dokument werden die Informationen zu den Grafiken hinterlegt. Der Adapter gibt ein PDF-Dokument aus, das um die angegebenen Grafiken ergänzt wurde, als Binärdatenstrom oder als XML-Dokument.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *AddImages*: Dem PDF-Dokument Grafiken aus mehrseitigen TIFF-Dateien hinzufügen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.pdf.PDFTiffImage* : Hauptklasse (Standard)
ReadPartially	PDF-Dokument in mehreren Teilen in den Arbeitsspeicher laden (kann Performance-Verbesserungen bringen) **Mögliche Werte:** • *ja*: In mehreren Teilen nacheinander in den Arbeitsspeicher laden • *nein*: PDF-Dokument(e) In einem Stück in den Arbeitsspeicher laden (Standard)
ToXML	PDF-Dokument Base64-kodiert in einem XML-Dokument ausgeben **Mögliche Werte:** • *ja*: XML-Dokument ausgeben • *nein*: PDF-Dokument als Binärdatenstrom ausgeben (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Funktions-Adapter erwartet eine adapterspezifische Input-XML-Struktur, in der Informationen zu jeder Grafik hinterlegt sind. Hierzu zählen Breite und Höhe (in *cm* oder *pt*), Drehung, x- und y-Position (Abstände vom unteren linken Seitenrand in *cm* oder *pt*), die Seite im Multipage-TIFF, der URL zur Grafik sowie die Seite im PDF-Dokument, auf der die Grafik positioniert werden soll.

Jedes Element `<TIFF/>` referenziert eine TIFF-Datei und jedes Element `<image/>` platziert eine Abbildung im PDF. Wenn es sich bei einem Input-PDF-Dokument um eine Base64-kodierte Daten handelt, muss stattdessen der auskommentierte Code angepasst und einkommentiert werden.

```
<pdfTiffImage>
   <PDF url="xstore-URL des Input-PDF-Dokuments im Repository"/>
   <!-- <PDF encoding="base64">(Wenn Input-PDF Base64-Zeichenkette)</PDF> -->

   <TIFF url="xstore-URL der TIFF-Grafik im Repository">
      <image index="Seite im Multipage-TIFF" page="Seite im PDF"
             height="Höhe" width="Breite" rad="Drehung (Bogenmaß)"
             degree="Drehung (Grad)" xPos="Abstand vom linken Seitenrand"
             yPos="Abstand vom unteren Seitenrand"/>
   </TIFF>
</pdfTiffImage>
```

Beispiel

Folgendes Input-XML-Dokument enthält Informationen zu zwei Seiten einer mehrseitigen TIF-Grafik, die in ein neues PDF-Dokument eingefügt werden sollen.

```
<pdfTiffImage>
   <PDF url="xstore://Project/Folder/Input.pdf"/>
   <!-- <PDF encoding="base64">base_64_encoded_PDF_document</PDF> -->

   <TIFF url="xstore://Project/Folder/Image.tif">
      <image index="1" page="1" height="100pt" width="300pt" rad=""
             degree="" xPos="9cm" yPos="26cm"/>
      <image index="3" page="2" height="" width="100pt" rad=""
             degree="" xPos="2.6cm" yPos="740pt"/>
   </TIFF>
</pdfTiffImage>
```

10.16.2.3 PDF to Image Converter

? Unknown Attachment `PDF to Image Converter`: Konvertiert und skaliert die erste Seite eines bestehenden, nicht geschützten PDF-Dokuments in ein Bildformat.

Folgende Formate können erzeugt werden: BMP, GIF, JPEG, JPEG 2000, PNG, PNM und TIFF. Der Adapter gibt die Grafik im gewünschten Dateiformat als Binärdatenstrom aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *Transform*: PDF in ein Bildformat konvertieren

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.pdf.PDF2Image*: Hauptklasse (Standard)
width	Breite des ausgegebenen Bildes in Pixel (wird ignoriert, wenn Parameter maxSize gesetzt ist) **Mögliche Werte:** • Beliebige positive ganze Zahl, z.B. *800* • *-1* (in Kombination mit einer positiven Pixelzahl in Parameter height): beim Skalieren des Bilds das ursprüngliche Seitenverhältnis beibehalten; in Kombination mit dem Wert -1 bei Parameter height werden die Originalbildabmessungen beibehalten
height	Höhe des ausgegebenen Bildes in Pixel (wird ignoriert, wenn Parameter maxSize gesetzt ist) **Mögliche Werte:** • Beliebige positive ganze Zahl, z.B. *600* • *-1* (in Kombination mit einer positiven Pixelzahl in Parameter `width`): beim Skalieren des Bilds das ursprüngliche Seitenverhältnis beibehalten; in Kombination mit dem Wert `-1` bei Parameter `width` werden die Originalbildabmessungen beibehalten
page	Seite innerhalb des PDF-Dokuments, die als Bild ausgegeben werden soll **Mögliche Werte:** • Beliebige ganze Zahl
maxSize	Maximale Breite oder Höhe des ausgegebenen Bildes in Pixel (Parameter `width` und `height` werden ignoriert) **Mögliche Werte:** • Beliebige ganze Zahl • *0*: Parameter `width` und `height` zur Skalierung verwenden (Standard)

scaleAlgorithm	Verwendeter Skalierungs-Algorithmus **Mögliche Werte:** • *`SCALE_DEFAULT`*: Standard-Algorithmus verwenden • *`SCALE_FAST`*: Schnellen Algorithmus verwenden • *`SCALE_SMOOTH`*: Kantenglättungs-Algorithmus verwenden • *`SCALE_REPLICATE`*: Pixelwiederholungs-Algorithmus verwenden • *`SCALE_AREA-AVERAGING`*: Flächen-Durchschnittsberechnungs-Algorithmus verwenden
outputFormat	Ausgegebenes Bildformat **Mögliche Werte:** • *`BMP`*: Windows-Bitmap-Grafik erzeugen • *`GIF`*: GIF-komprimiertes Bild erzeugen • *`JPEG`*: JPEG-komprimiertes Bild erzeugen • *`JPEG 2000`*: Bild im *JPEG 2000*-Format erzeugen • *`PNG`*: PNG-Bild erzeugen • *`PNM`*: Bild im *Portable Anymap*-Format erzeugen • *`TIFF`*: Bild im *Tagged Image File Format* erzeugen

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.16.2.4 PDF to TIFF Converter

? Unknown Attachment `PDF to TIFF Converter`: Konvertiert ein PDF-Dokument in eine TIFF-Grafik (*Tagged Image File Format*) und gibt die Grafik als Binärdatenstrom aus. Mehrseitige PDF-Dokumente werden in eine mehrseitige TIFF-Grafik (*Multipage TIFF*) konvertiert.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *`Transformation`*: PDF-Dokument in eine TIFF-Grafik konvertieren

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.pdf2tiff.PDFToTiffAdapter*: Hauptklasse (Standard)
compression	Kompressionstyp **Mögliche Werte:** • *Deflate (1)*: Per Deflate verlustfrei komprimieren • *JPEG (2)*: JPEG-Komprimierung (verlustbehaftet) verwenden • *None (3)*: Nicht komprimieren (Standard) • *Packbits (4)*: Per PackBits-Lauflängenkodierung verlustfrei komprimieren
deflateLvl	Deflate-Level (wenn Kompressionstyp *Deflate* gewählt wurde und die *Deflate*-Komprimierung auf die gesamte TIFF-Datei angewendet werden soll) **Mögliche Werte:** • Beliebige ganze Zahl von *1–9* (komprimiert bei *1* am schnellsten, bei *9* am ausführlichsten) • *0*: Keine *Deflate*-Komprimierung auf die gesamte TIFF-Grafik anwenden (Standard)
quality	Bei JPEG-Kompression: Qualitätsstufe der (verlustbehafteten) JPEG-Kompression **Mögliche Werte:** • Beliebige ganze Zahl von *0–100* (bei *0* geringste Qualität, bei *100* maximale Qualität) • *0*: Geringste Qualität, d.h. maximale Kompression (Standard)
resX	Horizontaler Bildschirm-/Druck-Auflösungswert **Mögliche Werte:** • Beliebige ganze Zahl (z. B. *72*) • *0*: Keinen Auflösungswert setzen

resY	Vertikaler Bildschirm-/Druck-Auflösungswert **Mögliche Werte:** • Beliebige ganze Zahl (z. B. *72*) • *0*: Keinen Auflösungswert setzen
resUnit	Einheit für die Bildschirm-/Druck-Auflösung **Mögliche Werte:** • *No absolute unit*: Keine absolute Einheit setzen • *Inch*: Zoll als Einheit verwenden (DPI bzw. PPI) • *Centimeter*: Zentimeter als Einheit verwenden

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Eine (optionale) CPU-Typ-/betriebssystemabhängige *JAI MediaLib*-Bibliothek wird nicht mitgeliefert. Der Adapter gibt deshalb folgende Fehlermeldung im Server-Protokoll aus, die für die Adapter-Ausführung jedoch nicht relevant ist:

```
ERROR [STDERR] Could not find mediaLib accelerator wrapper classes.
 Continuing in pure Java mode."
```

Um die *JAI MediaLib*-Bibliothek abzuschalten, folgende Java-VM-Option setzen (z. B. im Start-Skript):

```
@set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.media.jai.disableMediaLib=true
```

10.16.2.5 TIFF to PDF Converter

? Unknown Attachment `TIFF to PDF Converter`: Erzeugt aus einer mehrseitigen TIFF-Datei (*Multipage Tagged Image File Format*) ein PDF-Dokument im Format 210 x 297 mm (DIN A4), wobei jede Seite in der TIFF-Datei einer Seite im PDF-Dokument entspricht. Die Grafiken können im PDF-Dokument positioniert und anhand der Breite der Grafik proportional skaliert werden.

Beachten Sie, dass `.tif`-Dateien mit Alphakanälen nicht unterstützt werden.

Eigenschaften

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** `Convert`: PDF-Dokument aus einer mehrseitigen TIFF-Datei erstellen

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.pdf.Tiff2Pdf*: Hauptklasse (Standard)
`ToXML`	PDF-Dokument Base64-kodiert in einem XML-Dokument ausgeben **Mögliche Werte:** • `ja`: XML-Dokument ausgeben • `nein`: PDF-Dokument als Binärdatenstrom ausgeben (Standard)
`Header`	Text, der als Überschrift im PDF auf jeder Seite steht **Mögliche Werte:** Beliebige Zeichenkette
`xPos`	Abstand vom linken Seitenrand in pt **Mögliche Werte:** Beliebige ganze Zahl
`yPos`	Abstand vom unteren Seitenrand in pt **Mögliche Werte:** Beliebige ganze Zahl
`scale`	Grafiken skalieren und in die PDF-Seite einpassen **Mögliche Werte:** • `ja`: Grafiken skalieren • `nein`: Grafiken nicht skalieren, d.h. die in der TIFF-Datei festgelegte Auflösung verwenden (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Ausgabe

Der Adapter gibt ein PDF-Dokument als Binärdatenstrom oder als XML-Dokument aus, das alle Seiten der mehrseitigen TIFF-Datei enthält.

10.16.3 PDF Split and Merge

10.16.3.1 PDF Merger

`PDF Merger`: Fügt mehrere nicht geschützte PDF-Dokumente zu einem PDF-Dokument zusammen.

Dieser Adapter erwartet eine adapterspezifische Input-XML-Struktur, die mehrere Base64-kodierte Zeichenketten mit den Daten von PDF-Dokumenten enthält. Der Adapter gibt ein XML-Dokument aus, das eine Base64-kodierte Zeichenkette mit den Daten des zusammengefügten PDF-Dokuments enthält.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`Merge`*: Mehrere PDF-Dokumente zu einem PDF-Dokument zusammenfügen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.PDFMerger`*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Funktions-Adapter erwartet eine adapterspezifische Input-XML-Struktur, in der sämtliche Daten der PDF-Dokumente als Base64-kodierte Zeichenketten in der gewünschten Reihenfolge (festgelegt im Attribut `order`) hinterlegt sind.

```
<Data>
   <input order="1">
      <!--Base64-kodierte Zeichenkette mit Daten des ersten PDF-Dokuments-->
   </input>
   <input order="2">
      <!--Base64-kodierte Zeichenkette mit Daten des zweiten PDF-Dokuments-->
   </input>
</Data>
```

Ausgabe

Der erfolgreich ausgeführte Adapter gibt ein XML-Dokument mit folgendem Muster zurück:

```
<Data>
   <output mimetype="application/pdf" encoding="base64">
      <!--Base64-kodierte Zeichenkette mit Daten des zusammengefügten PDF-Dokuments--
>
   </output>
</Data>
```

10.16.3.2 PDF Page Counter

? Unknown Attachment `PDF Page Counter`: Ermittelt die Anzahl der Seiten eines PDF-Dokuments, das als Input-Dokument verarbeitet wird, und gibt ein XML-Dokument mit der Seitenzahl aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`GetPageCount`*: Anzahl der Seiten im PDF ermitteln (Standard)

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.pdf.PDFPageCount`*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Ausgabe

Der Funktions-Adapter gibt ein XML-Dokument mit folgender Struktur aus:

```
<Pages><!--Anzahl der Seiten--></Pages>
```

10.16.3.3 PDF Recombiner

`PDF Recombiner`: Fügt Seiten von mehreren nicht geschützten PDF-Dokumenten in einem neuen PDF-Dokument zusammen.

Dieser Adapter erwartet eine adapterspezifische Input-XML-Struktur, die sämtliche Verarbeitungs-Informationen für die PDF-Dokumente enthält. Der Adapter gibt die Daten eines zusammengefügten PDF-Dokuments als Binärdatenstrom oder als XML-Dokument aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *RecombinePDF*: Seiten von verschiedenen PDF-Dokumenten zu einem neuen PDF-Dokument zusammenfügen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.pdf.PDFRecombiner*: Hauptklasse (Standard)
ReadPartially	PDF-Dokument in mehreren Teilen in den Arbeitsspeicher laden (kann Performance-Verbesserungen bringen) **Mögliche Werte:** • *ja*: In mehreren Teilen nacheinander in den Arbeitsspeicher laden • *nein*: PDF-Dokument(e) In einem Stück in den Arbeitsspeicher laden (Standard)
ToXML	PDF-Dokument Base64-kodiert in einem XML-Dokument ausgeben **Mögliche Werte:** • *ja*: XML-Dokument ausgeben • *nein*: PDF-Dokument als Binärdatenstrom ausgeben (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Adapter erwartet eine adapterspezifische Input-XML-Struktur, die sämtliche Verarbeitungs-Informationen für das erzeugte PDF-Dokument enthält.

```
<pdfRecombiner>
   <PDF name="Deklaration eines Input-Dokumentbezeichners"
        url="xstore-URL eines Input-PDF-Dokuments im Repository"/>
   <!-- <PDF name="inputname"
             encoding="base64">(Wenn Input-PDF Base64-Zeichenkette)
        </PDF> -->

   <Range pdf="Input-Dokumentbezeichner" from="Startseite"
          to="Endseite" odd="ungerade Seiten (true/false)"
          even="gerade Seiten (true/false)"/>
   <Page pdf="Input-Dokumentbezeichner"
         pageNr="Einzelne Seitenzahl"/>
   <Empty count="Anzahl leerer Seiten"/>
   <All pdf="Input-Dokumentbezeichner"/>
</pdfRecombiner>
```

Erläuterung:

- `<PDF/>`: referenziert je ein Input-Dokument.
  Wenn es sich bei einem Input-PDF-Dokument um Base64-kodierte Daten handelt, muss stattdessen der auskommentierte Code angepasst und einkommentiert werden.
- `<Range/>` definiert einen Seitenbereich von Start- bis Endseite.
- `<Page/>` definiert eine Einzelseite.
- `<All/>` definiert sämtliche Seiten eines Input-PDF-Dokuments.
- `<Empty/>` definiert leere Seiten im erzeugten PDF-Dokument.

Beispiel

Folgendes Input-XML-Dokument enthält Informationen zu zwei PDF-Dokumenten, die mit den definierten Seiten und insgesamt drei Leerseiten in einem neuen PDF-Dokument zusammengefügt werden sollen.

```
<pdfRecombiner>
   <PDF name="input1" url="xstore://Project/Folder/InputPDF1.pdf"/>
   <PDF name="input2" url="xstore://Project/Folder/InputPDF2.pdf"/>
   <!-- <PDF name="inputname"
             encoding="base64">base_64_encoded_PDF_document</PDF> -->

   <Range pdf="input1" from="1" to="10" odd="false" even="true"/>
   <Page pdf="input1" pageNr="1"/>
   <Empty count="2"/>
   <Range pdf="input2" from="5" to="10"/>
   <Empty count="1"/>
   <All pdf="input1"/>
</pdfRecombiner>
```

10.16.3.4 PDF Splitter Iterator

`PDF Splitter Iterator`: Teilt ein nicht geschütztes PDF-Dokument in mehrere neue PDF-Dokumente.
Dieser Adapter erwartet eine adapterspezifische Input-XML-Struktur, die den Pfad zum verarbeiteten PDF-Dokument im Repository enthält. Der Adapter gibt je Iteration die Daten eines PDF-Teildokuments als Binärdatenstrom oder als XML-Dokument aus.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • `Splitt Iterator`: PDF-Dokument in mehrere PDF-Dokumente teilen • `Clear`: Iterator auf Anfangswert zurücksetzen

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** `de.softproject.integration.adapter.pdf.PDFSplitterIteratorAdapter`: Hauptklasse (Standard)
`Start`	Seitenzahl, von der gestartet werden soll **Mögliche Werte:** Beliebige positive ganze Zahl
`Step`	Schrittgröße, um die der Parameter `Start` für die nächste Iteration hochgezählt wird. **Mögliche Werte:** Beliebige positive ganze Zahl
`Count`	Anzahl der Seiten, die pro Iteration als separates PDF-Dokument ausgegeben werden sollen **Mögliche Werte:** Beliebige positive ganze Zahl
`ToXML`	PDF-Dokument Base64-kodiert in einem XML-Dokument ausgeben **Mögliche Werte:** • `ja`: XML-Dokument ausgeben • `nein`: PDF-Dokument als Binärdatenstrom ausgeben (Standard)

`ReadPartially`	PDF-Dokument in mehreren Teilen in den Arbeitsspeicher laden (kann Performance-Verbesserungen bringen) **Mögliche Werte:** • `ja`: In mehreren Teilen nacheinander in den Arbeitsspeicher laden • `nein`: PDF-Dokument(e) in einem Stück in den Arbeitsspeicher laden (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Funktions-Adapter erwartet eine adapterspezifische Input-XML-Struktur, die den Pfad zum verarbeiteten PDF-Dokument im Repository oder die Base64-Zeichenkette enthält.

PDF im PDF-Format

Folgendes Input-XML-Dokument enthält den Pfad zu einem PDF-Dokument im Repository.

```
<?xml version="1.0" encoding="UTF-8"?>
<PDF url="xstore://Project/Folder/Input.pdf"/>
```

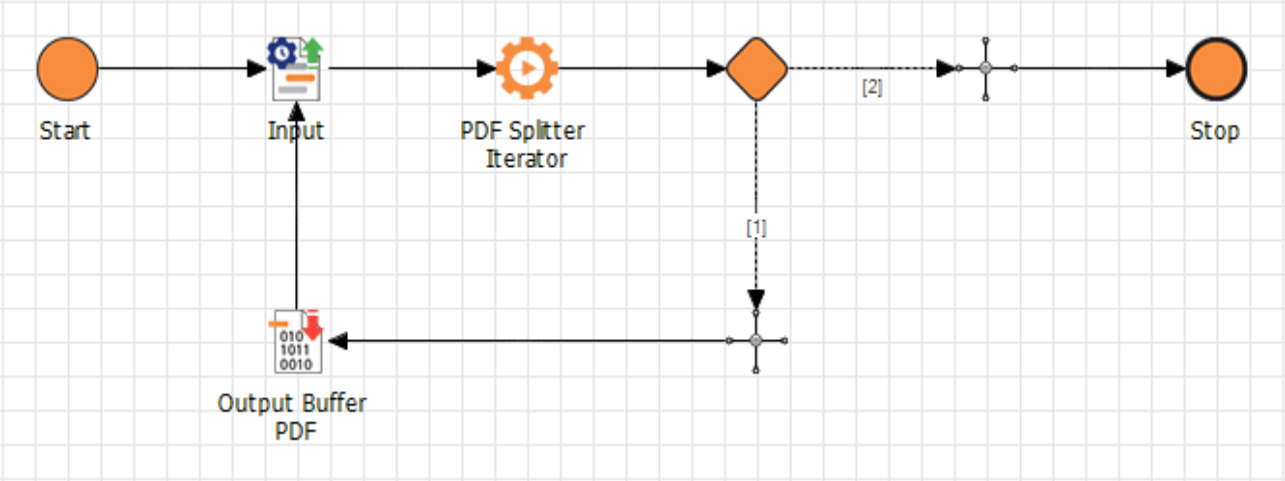
PDF im Base64-Format

Folgendes Input-XML-Dokument enthält die Base64-Zeichenkette.

```
<?xml version="1.0" encoding="UTF-8"?>
<PDF encoding="base64">
  Base64-Zeichenkette
</PDF>
```

Output

Wenn der Parameter `ToXML` **deaktiviert** ist, kann der Output direkt als Buffer-Datei im PDF-Format gespeichert werden.

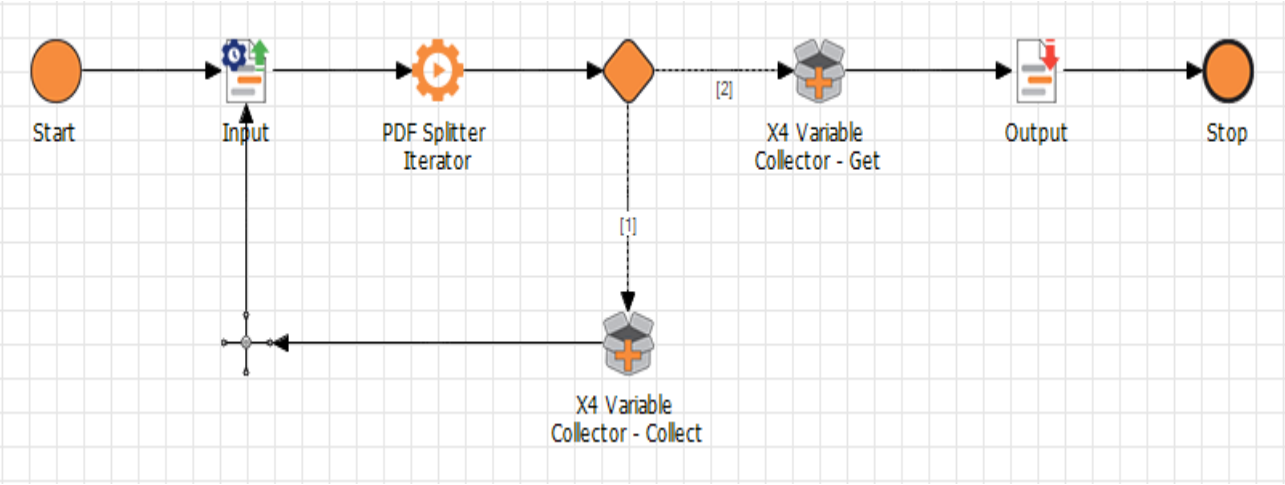


Wenn der Parameter `ToXML` **aktiviert** ist, sieht der Output wie folgt aus:

```
<?xml version="1.0" encoding="UTF-8"?>
<PDF encoding="base64">
    Base64-Zeichenkette
</PDF>
```

10.16.3.4.1 Beispiel

Der Prozess teilt die XML auf und speichert alle Zwischenergebnisse im XML-Format in einer Variablen. Um die Variablen zu speichern, wird der Adapter X4 Variable Collector verwendet.



10.17 Security

In Kategorie *Security* werden Adapter und Konnektoren zur Verfügung gestellt, die es ermöglichen, Daten in Technical Processes zu verschlüsseln und zu entschlüsseln sowie Signaturen zu erzeugen und zu validieren.

10.17.1 Antivirus

`Antivirus`: Stellt eine Verbindung zu einem Server der ClamAV, einer Open Source (*GPL*) Antivirus-Engine, her. Eingabedateien werden zur Laufzeit des Adapters auf Viren überprüft.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *Scan*: Eingabedateien auf Viren überprüfen.

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** `de.softproject.integration.adapter.antivirus.AntiVirusAdapter`: Hauptklasse (Standard)
host	IP-Adresse oder Host-Name des ClamAV-Servers **Mögliche Werte:** • Beliebige gültige Zeichenkette, z. B. `localhost`, `127.0.0.1` • *`localhost`*: ClamAV Host (Standard)
port	Portnummer des ClamAV-Servers **Mögliche Werte:** • Jede beliebige Ganzzahl (Integer) • *`3310`*: FTP-Port (Standard)
scanTimeoutInMs	Wartezeit auf einen Scan in Millisekunden. **Mögliche Werte:** • Jede beliebige Ganzzahl (Integer). Der Standardwert ist 500. • Bei einem Wert von 0 wird kein Timeout gesetzt. Das kann zu unendlichen Wartezeiten führen. • Negative Werte werden ignoriert, und der Standardwert 500 wird verwendet.

Statuswerte

`1`	Der Adapter gibt ein Ergebnisdokument aus. Die Operation war erfolgreich, es wurden keine Viren gefunden *oder* die Eingabedatei ist leer.
`-1`	Der Adapter gibt die Eingabedatei aus. Die Operation konnte nicht ausgeführt werden. Details zur Fehlermeldung können den Server-Logs entnommen werden.
`400`	Der Adapter gibt ein Ergebnisdokument aus. Die Operation war erfolgreich, es wurden Viren in der Eingabedatei gefunden.

Input

Der Adapter erwartet jede Eingabedatei, sowohl Binärdateien als auch Base64-kodierte Dokumente (Text- oder XML-Dateien).

Ausgabe

Wenn die Operation fehlschlug, gibt der Adapter die Eingabedatei aus.

Wenn die Operation erfolgreich war, gibt der Adapter ein UTF-8-kodiertes XML-Ergebnisdokument der nachfolgenden Struktur aus. Wurden Viren gefunden, wird der Wert auf `true` gesetzt, sonst wird der Wert `false` ausgegeben.

```
<?xml version="1.0" encoding="UTF-8"?>
<ScanResult>
    <!--Der Wert true wird ausgegeben, wenn in der Eingabedatei Viren gefunden
wurden-->
    <!--Der Wert false wird ausgegeben, wenn in der Eingabedatei keine Viren gefunden
wurden-->
    <VirusFound><!--Wert--></VirusFound>
</ScanResult>
```

Abhängigkeiten und Installationshinweise

Dieser Adapter verwendet folgende Bibliotheken: ClamAV Bibliothek `clamav-client-1.0.1.jar`

10.17.1.1 Installation ClamAV

Für eine einwandfreie Funktion des Adapters muss der ClamAV-Server auf derselben Maschine wie die X4 BPMS installiert sein und laufen. Eine ausführliche Installationsbeschreibung und Anpassungen an den Konfigurationsdateien finden Sie in dem offiziellen ClamAV Handbuch (*siehe Kapitel 4 und 5*).

Download Intallationspakete
Eine Auflistung der verfügbaren Installationspakete finden Sie auf der Website des Anbieters: https://www.clamav.net/downloads#otherversions

- Download-Version mit Installationsassistent für Windows (32 bit): `clamav-<version_number>-x86.zip`
- Download-Version mit Installationsassistent für Windows (64 bit): `clamav-<version-number>-win-x64.zip`

10.17.1.2 Aktualisierung der Virendatenbank

Die Einstellungen zur automatischen Aktualisierung der Virendatenbank sind in dem offziellen ClamAV Handbuch (*siehe Kapitel 4)* beschrieben.

10.17.2 BCrypt Calculator

`BCrypt Calculator`: Erstellt und verifiziert Salted BCrypt-Hashes.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `GenerateHash`: erstellt einen Salted Hash zum eingegebenen Klartext-Passwort • `Verify`: verifiziert das Klartext-Passwort mit dem eingegebenen Salted Hash • `VerifyAndReplace`: verifiziert das Klartext-Passwort mit dem eingegebenen Salted Hash und erstellt einen neuen Salted Hash

Parameter

Adapter	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.bcrypt.BCryptAdapter*: Hauptklasse (Standard)
`clearText`	Klartext-Passwort **Mögliche Werte:** Beliebiger String, z. B. *`MeinPasswort`* Der Parameter darf nicht leer sein.

`bcryptHash`	Salted BCrypt-Hash, der mit dem Klartext-Passwort verglichen werden soll; Nötig bei den Operationen `Verify` und `VerifyAndReplace` **Mögliche Werte:** Beliebiger Salted BCrypt-Hash
`newclearText`	Neues Klartext-Passwort; Nur nötig bei der Operation `VerifyAndReplace` **Mögliche Werte:** Beliebiger String, z. B. *`MeinPasswort`* Der Parameter darf nicht leer sein.

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt und im Output befindet sich ein XML-Dokument mit dem Resultat
`0`	Operationen `Verify` und `VerifyAndReplace`: der Parameter `bcryptHash` stimmt nicht mit dem `clearText`-String überein
`-1`	Es kam zu einem Fehler während der Ausführung • Die erforderlichen Parameter sind nicht befüllt • Operationen `Verify` und `VerifyAndReplace`: der Parameter `bcryptHash` ist nicht mit einem Salted BCrypt-Hash befüllt

Input

Der Adapter erwartet bei keiner Operation einen Input.

Ausgabe

Je nach Operation gibt der Adapter verschiedene Ergebnisse aus:

- Operationen `GenerateHash` und `VerifyAndReplace`: XML-Dokument mit dem Salted Hash
- Operation `Verify`: XML-Dokument mit dem Ergebnis, ob der Salted Hash dem Klartext-Passwort entspricht
- Operation `VerifyAndReplace` (wenn Salted Hash nicht mit Klartext-Passwort übereinstimmt): XML-Dokument mit dem Ergebnis, dass der Salted Hash nicht dem Klartext-Passwort entspricht
- XML-Dokument mit Fehlermeldungen

Beispiel-Ausgabe

```
<?xml version="1.0" encoding="UTF-8"
<Hash>$2a$10$EcgngKl9hSlppvEcDspLzeq.9Blqim1SFNVzY6tbn5ixTZFMh4xs</Hash>
```

10.17.3 Checksum Creator

? Unknown Attachment `Checksum Creator`: Bildet Prüfsummen für Dateien in einem Ordner und gibt die Prüfsummen gemeinsam mit den Dateinamen und Datei-Eigenschaften als XML-Dokument aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`Count CRC`*: CRC-Algorithmus zum Bilden der Prüfsummen verwenden • *`Count Adler`*: Adler-Algorithmus zum Bilden der Prüfsummen verwenden

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.checksum.ChecksumAdapter`*: Hauptklasse (Standard)
path	Ordner, für dessen Inhalt Prüfsummen erstellt werden soll **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *`%USER_COLLECTION%/Project/Folder`*)
useRecursion	Ordner rekursiv behandeln **Mögliche Werte:** • *`ja`*: Elemente in Unterordnern einbeziehen • *`nein`*: Auf die Elemente im angegebenen Ordner beschränken

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Adapter benötigt keinen Input aus dem vorhergehenden Prozess-Schritt.

Ausgabe

Beispiel für eine ausgegebenes XML-Dokument mit Dateinamen, Prüfsummen und einem einbezogenen Unterordner:

```
<list>
    <directory lastModifiedAsString="30.09.2009 15:22:47 MESZ"
       lastModifiedAsLong="1254316967185" size="0" hidden="false">Folder
    </directory>
    <file lastModifiedAsString="30.09.2009 15:22:47 MESZ"
       lastModifiedAsLong="1254316967185" size="123" hidden="false"
       checksum="1376899901">Folder\File.txt</file>
    <file lastModifiedAsString="30.09.2009 15:22:27 MESZ"
       lastModifiedAsLong="1254316947494" size="6" hidden="false"
       checksum="63888547">Textfile.txt</file>
    <file lastModifiedAsString="30.09.2009 15:22:39 MESZ"
       lastModifiedAsLong="1254316959989" size="6" hidden="false"
       checksum="4270080073">XMLDocument.xml</file>
</list>
```

10.17.4 Digest Calculator

`Digest Calculator`: Generiert bzw. berechnet den Hash-Code (RSA, MD5 etc.) der Input-Daten.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *`Calculate`*: Hash-Code berechnen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.digest.DigestAdapter`*: Hauptklasse (Standard)
algorithm	Verwendeter Algorithmus zur Berechnung des Hashcodes **Mögliche Werte:** • *`MD5`*: *Message-Digest 5*-Algorithmus verwenden (Standard) • *`SHA`*: *Secure Hash (0)*-Algorithmus verwenden • *`SHA-1`*: *Secure Hash 1*-Algorithmus verwenden • *`SHA-256`*: *Secure Hash 256*-Algorithmus verwenden • *`CRC`*: *Cyclic Redundancy Check*-Algorithmus verwenden • *`ADLER`*: *Adler*-Algorithmus verwenden

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Ausgabe

Dieser Adapter gibt ein *UTF-8* -kodiertes XML-Dokument aus, das den verwendeten Algorithmus und den berechneten Hashcode enthält:

```
<digest type="Algorithmus"><!--Hashcode der Input-Daten--></digest>
```

Der Adapter gibt beispielsweise folgendes XML-Dokument aus:

```
<digest type="MD5">2e5941901e8b5e214dcc5032dfce71bb</digest>
```

10.17.5 JSON WebToken Calculator

`JSON WebToken Calculator`: Generiert, verifiziert und verarbeitet JSON-WebTokens.

Name des Adapters bis Version 5.5.4: *JSON WebToken*

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `CreateJWT`: Generiert ein JSON-WebToken (JWT) aus einem XML-Dokument • `Verify`: Verifiziert den vom X4 Server generierten JWT • `CreateXML`: Erstellt aus einem vom X4 Server generierten JWT das dazugehörige XML-Dokument

Parameter

Folgende Parameter sind bei den oben genannten Operationen erforderlich:

	CreateJWT	Verify	CreateXML
algorithm	erforderlich	erforderlich	erforderlich
useServerSecret		erforderlich, falls `useSecretAsPublicKey` nicht verwendet wird.	
useSecretAsPublicKey		erforderlich, falls `useServerSecret` nicht verwendet wird.	
secret		erforderlich, falls `useSecretAsPublicKey` verwendet wird.	
keystoreUrl	erforderlich		erforderlich

keystoreType	erforderlich		erforderlich
keystorePassword	erforderlich		erforderlich
keyAlias	erforderlich		erforderlich
keyPassword	erforderlich		erforderlich

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.json.webtokenJSONWebTokenAdapter*: Hauptklasse (Standard)
`algorithm`	Algorithmus, der für die Operation verwendet wird **Mögliche Werte:** • `HMAC256` • `HMAC384` • `HMAC512` • `RSA256` • `RSA384` • `RSA512` • `ECDSA256` • `ECDSA384` • `ECDSA512`
`useServerSecret`	*Nur relevant für HMAC-Algorithmen.* Legt fest, ob ein integriertes Server Secret oder eine benutzerdefiniertes Server Secret verwendet wird. **Mögliche Werte:** • `true`: Integriertes Server Secret verwenden • `false`: Benutzerdefiniertes Server Secret verwenden ⚠ Falls ein benutzerdefiniertes Server Secret verwendet wird, muss dieses mit dem Parameter `secret` angegeben werden.
`useSecretAsPublicKey`	*Nur relevant für RSA- und ECDSA-Algorithmen.* Legt fest, ob der Wert des Parameters `secret` als Public Key verwendet wird. **Mögliche Werte:** • `true`: Wert des Parameters `secret` als Public Key verwenden ⚠ Falls der Wert des Parameters `secret` als Public Key verwendet wird, dann ist der Parameter `secret` erforderlich. • `false`: Wert des Parameters `secret` nicht als Public Key verwenden

`secret`	*Nur relevant für HMAC-Algorithmen.* Kennwort, dass verwendet wird, um ein Token zu verschlüsseln oder zu entschlüsseln. **Mögliche Werte:** Base64-verschlüsseltes Kennwort
`keystoreUrl`	URL zum Keystore, der das RSA-/ECDSA-Schlüsselpaar enthält **Mögliche Werte:** URL zu einem Keystore ⚠ Der Keystore muss durch die Java Security API unterstützt werden, z. B. JKS oder PKCS#12.
`keystoreType`	Typ des Keystores, der das RSA-/ECDSA-Schlüsselpaar enthält **Mögliche Werte:** Typ des Keystores, z. B. `jks` oder `pkcs12`
`keystorePassword`	Passwort des Keystores, der das RSA-/ECDSA-Schlüsselpaar enthält **Mögliche Werte:** Passwort
`keyAlias`	Alias des asymmetrischen Schlüssels, der für RSA-/ECDSA-Algorithmen verwendet wird **Mögliche Werte:** Alias des verwendeten Schlüssels
`keyPassword`	Passwort des asymmetrischen Schlüssels, der für RSA-/ECDSA-Algorithmen verwendet wird **Mögliche Werte:** Passwort des verwendeten Schlüssels

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt: • Operation `CreateJWT`: Im Output befindet sich ein JSON-WebToken • Operation `Verify`: Der JSON-WebToken ist gültig • Operation `CreateXML`: Im Output befindet sich das XML-Dokument zum JSON-WebToken
`-1`	Bei der Ausführung des Adapters kam es zu einem Fehler: • Operation `CreateJWT` und `CreateXML`: Im Output befindet sich eine Fehlermeldung • Operation `Verify`: Der JSON-WebToken ist ungültig.
`400`	Das Token ist abgelaufen. ⓘ Dieser Statuswert wird nur von den Operationen `CreateXML` und `Verify` ausgegeben.

Input

Je nach Operation erwartet der Adapter verschiedene Inputs:

- Operation `CreateJWT`: es wird ein XML-Dokument erwartet, welches folgendes Format besitzt:

⚠
- Die Header-Typen `typ` (Parameter: `keystoretype`) und `alg` (Parameter: `algorithm`) wird vom Adapter gesetzt.
- Verwenden Sie keine Claims mit duplizierten Namen.

ⓘ Das Element Header ist optional und kann bei Bedarf deklariert werden. Zum Beispiel erwartet Microsoft Azure eine Header-Deklaration.

Beispiel-Input

```
<?xml version="1.0" encoding="UTF-8">
<TokenContent>
    <Headers>
        <Header name="x5t">hOBcHZi846VCHSJbFAs26Go9VTQ=</Header>
        <Header name="kid">vdgdINF455FBKWDSEF5gdrgddgg=</Header>
    </Headers>
    <Subject name="sub" type="String">1234567890</Subject>
    <ExpiresAt name="exp" type="Integer">1577750400</ExpiresAt>
    <NotBefore name="nbf" type="Integer">946684800</NotBefore>
    <Id name="jti" type="String">007</Id>
    <Claims>
        <Claim name="name" type="String">John Doe</Claim>
        <Claim name="admin" type="boolean">true</Claim>
    </Claims>
</TokenContent>
```

ⓘ Zusätzliche Informationen zu den verwendenden Claims und deren XML-Objekten finden Sie unter iana.org. Bei nicht reservierten Claims können die Namen beliebig vergeben werden, jedoch dürfen sie nicht gleich sein mit bereits reservierten Claims wie beispielsweise *bsp.* oder *exp.*

ⓘ Als `type` stehen die Werte *String*, *Boolean*, *Integer*, *Long* und *Double* zur Verfügung.

ⓘ Für die Elemente `ExpiresAt`, `NotBefore` und `IssuedAt` werden Angaben im Unix-Zeitformat erwartet.

- Operationen `Verify` und `CreateXML`: Es wird ein JSON-WebToken erwartet, das aus einem Zwischenschritt stammen oder als Text-Datei (`.txt`) übergeben werden kann.

Ausgabe

Je nach Operation gibt der Adapter verschiedene Ergebnisse aus:

- Operation `CreateJWT`: Im Output befindet sich ein JSON-WebToken
- Operation `Verify`: Wenn der JWT gültig ist, wird er wieder ausgegeben
- Operation `CreateXML`: Im Output befindet sich das XML-Dokument zum JSON-WebToken

10.17.6 LDAP Connector

`LDAP Connector`: Stellt eine Verbindung zu einem LDAP-Server (*Lightweight Directory Access Protocol*) her, um Kontaktdaten abzufragen – bei Bedarf über LDAP Secure. Über adapterspezifische XML-Dokumente als Input-Daten können verschiedene Operationen ausgeführt werden.

> Die Konfiguration für LDAP Secure mit selbstsignierten Zertifikaten lässt sich in der Konfigurationsdatei `X4config.xml` vornehmen, siehe LDAPS-Konfiguration.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *`Execute`*: Adapter ausführen

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.ldap.Ldap`*: Hauptklasse (Standard)
`java.naming.factory.initial`	Name der Klasse, die den LDAP-Dienst bereitstellt **Mögliche Werte:** Beliebiger Klassenname, z.B. *`com.sun.jndi.ldap.LdapCtxFactory`*
`java.naming.provider.url`	URL des LDAP(S)-Servers **Mögliche Werte:** Beliebige URL
`java.naming.security.authentication`	Authentifizierungsmethode **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *`simple`*)

`java.naming.security.principal`	Name des Benutzers bzw. der Authentifizierungs-Anwendung (abhängig von der Authentifizierungsmethode) **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *`cn=S. User, ou=NewHires, o=JNDITutorial`*)
`java.naming.security.credentials`	Zugehöriges Passwort des Benutzers bzw. der Authentifizierungs-Anwendung (abhängig von der Authentifizierungsart) **Mögliche Werte:** Beliebige Zeichenkette (z.B. *`mysecret`*)
`java.naming.referral`	Definiert, wie mit Weiterleitungen umgegangen werden soll. **Mögliche Werte:** • `ignore`: Weiterleitungen ignorieren • `follow`: Allen Weiterleitungen automatisch folgen • `throw`: Für jede Weiterleitung eine ReferralException auslösen (siehe ReferralException auslösen)
`attributesAsElements`	Attribute eines Objektes ausgeben **Mögliche Werte:** • `true`: Attribute eines Objektes ausgeben • `false`: Keine Objekt-Attribute ausgeben (Standard)
`acceptAllCertificates`	Deaktiviert die Sicherheitseinstellungen durch Akzeptieren eines beliebigen Zertifikats **Mögliche Werte:** • `true`: Ein beliebiges Zertifikat wird akzeptiert • `false`: Nur selbstsignierte Zertifikate werden akzeptiert (Standard)
`acceptAllHostnames`	Deaktiviert die Sicherheitseinstellungen durch Akzeptieren eines beliebigen im Zertifikat enthaltenen Hostnamen. **Mögliche Werte:** • `true`: Akzeptiert einen beliebigen Hostnamen • `false` : Sicherheitseinstellungen sind aktiviert (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.

`2`	*Operation `Compare` erforderlich.* LDAP-Eintrag stimmt mit den Angaben im Input-XML-Dokument überein.
`3`	*Operation `Compare` erforderlich.* LDAP-Eintrag stimmt **nicht** mit den Angaben im Input-XML-Dokument überein.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

ReferralException auslösen

Wenn Weiterleitungen manuell verarbeitet werden sollen, dann müssen folgende Schritte befolgt werden:

1. Exception auffangen.
2. Weiterleitungsinformationen auslesen, indem `ReferralException.getReferralInfo()` verwendet wird und z. B. den Benutzer fragen, ob der Weiterleitung gefolgt werden soll.
   a. Wenn der Weiterleitung gefolgt werden soll, den Weiterleitungskontext mit `ReferralException.getReferralContext()` auslesen und die ursprüngliche Kontextmethode mit den gleichen Argumenten erneut aufrufen, die auch für den ursprünglichen Aufruf verwendet werden sollte.
   b. Wenn der Weiterleitung nicht gefolgt werden soll, `ReferralException.skipReferral()` aufrufen.
      i. Wenn diese Methode `true` zurückgibt, dann bedeutet das, dass mehreren Weiterleitungen gefolgt werden soll.
      Wenn das der Fall ist, `ReferralException.getReferralContext()` aufrufen, um fortzufahren. Wenn eine Kontextmethode das Ergebnis aufruft, dann wird für die nächste Weiterleitung wieder eine Exception ausgelöst. Diese Exception ab Schritt 1 behandeln.
      ii. Wenn die Methode `false` zurückgibt, dann gibt es keine Verweise mehr und dieses Verfahren kann beendet werden.

Weitere Informationen unter https://docs.oracle.com/javase/jndi/tutorial/ldap/referral/throw.html.

Input

Dieser Funktions-Adapter erwartet spezifische Input-XML-Strukturen, die LDAP-Befehle enthalten. Diese Input-XML-Strukturen können über XSL-Mappings in Abhängigkeit von den verarbeiteten Daten dynamisch generiert werden.

10.17.6.1 search-Befehl

Mit dem LDAP-Befehl `search` können Sie Einträge im LDAP-Verzeichnis suchen.

Input

Der *LDAP Adapter* erwartet zum Ausführen des `search`-Befehls folgende Input-XML-Struktur:

```
<?xml version="1.0" encoding="UTF-8"?>
<ldap>
    <search>
        <!-- Distinguish Name - base node in LDAP tree where to start search -->
        <name>OU=Accounts,DC=softproject,DC=local</name>
        <!-- Search filter - required -->
        <!-- filter that is applied on search result -->
        <filter>objectclass=user</filter>
        <!-- Search control settings -->
        <controls>
            <!-- Search scope. -->
            <!-- Keywords: object, subtree, onelevel -->
            <!-- object - Results from search will contain one or zero element. -->
            <!-- One element if the named object satisfies the search -->
            <!-- filter specified in search. -->
            <!-- subtree - Search the entire subtree rooted at the named object. -->
            <!-- onelevel - Search one level of the named context. -->
            <scope>subtree</scope>
            <!-- Time limit (ms) - for request duration -->
            <timeLimit>5000</timeLimit>
            <!-- List of attributes that should be returned, separated by colon. -->
            <attributes>uSNCreated,givenname,displayName,cn</attributes>
            <!-- Count limitation for result -->
            <countLimit>100</countLimit>
        </controls>
    </search>
</ldap>
```

Ausgabe

Der erfolgreich ausgeführte Adapter gibt folgendes XML-Dokument aus:

```
 <ldap name="dc=sp-linux" filter="(objectclass=*)" scope="subtree"
    attributes="cn,sn" countLimit="4">
    <entry dn="dc=sp-linux" />
    <entry dn="cn=Manager,dc=sp-linux">
        <cn>Manager</cn>
    </entry>
    <entry dn="o=softproject,dc=sp-linux" />
    <entry dn="sn=Tester,o=SoftProject,dc=sp-linux">
        <sn>Tester</sn>
        <cn>John</cn>
    </entry>
    </ldap>
```

10.17.6.2 add-Befehl

Mit dem LDAP-Befehl add können Sie Einträge dem LDAP-Verzeichnis hinzufügen.

Input

Der *LDAP Adapter* erwartet zum Ausführen des add-Befehls folgende Input-XML-Struktur:

```
<ldap>
    <add>
        <name>sn=Tester, o=SoftProject, dc=sp-linux</name>
        <!--dn-->
        <attributes>
            <objectClass>person</objectClass>
            <sn>Tester</sn>
            <cn>John</cn>
        </attributes>
        <!--list of attributes that will be added-->
    </add>
    </ldap>
```

Ausgabe

Der erfolgreich ausgeführte Funktions-Adapter gibt nach der Ausführung des `add`-Befehls kein XML-Dokument aus, sondern lediglich den Status `1 (successful)` oder im Fehlerfall den Status `-1 (Error)`.

10.17.6.3 modify-Befehl

Mit dem LDAP-Befehl `modify` können Sie Einträge im LDAP-Verzeichnis verändern. Dabei werden wie in einer Transaktion entweder alle Einträge geändert oder im Fehlerfall keine Änderungen an den Einträgen vorgenommen.

Input

Der *LDAP Adapter* erwartet zum Ausführen des `modify`-Befehls folgende Input-XML-Struktur:

```
<ldap>
    <modify>
        <name>dc=sp-linux</name>
        <!--dn-->
        <add>
            <o>Tester</o>
        </add>
        <replace>
            <o>SoftProject</o>
        </replace>
        <remove>
            <o>SoftProject</o>
        </remove>
    </modify>
    </ldap>
```

Beispiel für ein Passwort `replace`:

```
<ldap>
    <modify>
        <name>dc=sp-linux</name>
        <add>
            <o>Tester</o>
        </add>
        <replace>
            <unicodePW base64="true">dGVzdCE=</unicodePW>
        </replace>
    </modify>
</ldap>
```

Ausgabe

Der erfolgreich ausgeführte Funktions-Adapter gibt nach der Ausführung des `modify`-Befehls kein XML-Dokument aus, sondern lediglich den Status `1 (successful)` oder im Fehlerfall den Status `-1 (Error)`.

10.17.6.4 modifyDN-(RDN)-Befehl

Mit dem LDAP-Befehl `modifyDN (RDN)` können Sie Einträge im LDAP-Verzeichnis bearbeiten.

Input

Der *LDAP Adapter* erwartet zum Ausführen des `modifyDN-RDN`-Befehls folgende Input-XML-Struktur:

```
<ldap>
    <modifyDN-RDN>
        <oldName>sn=Tester, o=SoftProject, dc=sp-linux</oldName>
        <!--dn-->
        <newName>sn=TestUser, o=SoftProject, dc=sp-linux</newName>
        <!--dn-->
    </modifyDN-RDN>
    </ldap>
```

Ausgabe

Der erfolgreich ausgeführte Funktions-Adapter gibt nach der Ausführung des `modifyDN-RDN`-Befehls kein XML-Dokument aus, sondern lediglich den Status `1 (successful)` oder im Fehlerfall den Status `-1 (Error)`.

10.17.6.5 delete-Befehl

Mit dem LDAP-Befehl `delete` können Sie Einträge im LDAP-Verzeichnis löschen.

Input

Der *LDAP Adapter* erwartet zum Ausführen des `delete`-Befehls folgende Input-XML-Struktur:

```
<ldap>
    <delete>
        <name>sn=Tester, o=SoftProject, dc=sp-linux</name>
        <!--dn-->
    </delete>
</ldap>
```

Ausgabe

Der erfolgreich ausgeführte Funktions-Adapter gibt nach der Ausführung des `delete`-Befehls kein XML-Dokument aus, sondern lediglich den Status `1 (successful)` oder im Fehlerfall den Status `-1 (Error)`.

10.17.6.6 compare-Befehl

Mit dem LDAP-Befehl `compare` können Sie Einträge im LDAP-Verzeichnis vergleichen.

Input

Der *LDAP Adapter* erwartet zum Ausführen des `compare`-Befehls folgende Input-XML-Struktur:

```
<ldap>
    <compare>
        <name>sn=Tester, o=SoftProject, dc=sp-linux</name>
        <!--dn-->
        <attributes>
            <objectClass>person</objectClass>
            <sn>Tester</sn>
            <cn>John</cn>
        </attributes>
        <!--list of attributes that will be compared by values-->
        <controls>
            <scope>subtree</scope>
            <!--keywords: object,subtree,onelevel-->
            <timeLimit />
            <!--timeLimit (ms)-->
        </controls>
    </compare>
    </ldap>
```

Ausgabe

Der erfolgreich ausgeführte Funktions-Adapter gibt nach der Ausführung des `compare`-Befehls kein XML-Dokument aus, sondern lediglich den Status `CompareTRUE`, wenn der LDAP-Eintrag mit den Angaben im Input-XML-Dokument übereinstimmt, oder im andernfalls den Status `CompareFALSE`.

10.17.7 Password Generator

`Password Generator`: Erzeugt ein zufälliges Passwort anhand von Passwortrichtlinien, die in den Adapter-Parametern definiert werden. Dieser Adapter erwartet keinen Input und gibt ein XML-Dokument mit dem erzeugten Passwort aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** `Generate`: Ein Passwort erzeugen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.security.PasswordGenerator`*: Hauptklasse (Standard)
length	Anzahl der Zeichen für das zu erzeugende Passwort **Mögliche Werte:** • Beliebige positive ganze Zahl • `8`: Passwort mit 8 Zeichen erzeugen (Standard)
numberRequired	Definiert, ob eine nummerische Ziffer im Passwort enthalten sein soll **Mögliche Werte:** • *`True`*: Passwort wird eine nummerische Ziffer enthalten • *`False`*: Passwort wird keine nummerische Ziffer enthalten (Standard)
lowerCaseCharRequired	Definiert, ob ein Kleinbuchstabe im Passwort enthalten sein soll **Mögliche Werte:** • *`True`*: Passwort wird einen Kleinbuchstaben enthalten • *`False`*: Passwort wird keinen Kleinbuchstaben enthalten (Standard)

upperCaseCharRequired	Definiert, ob ein Großbuchstabe im Passwort enthalten sein soll **Mögliche Werte:** • *True*: Passwort wird einen Großbuchstaben enthalten • *False*: Passwort wird keinen Großbuchstaben enthalten (Standard)
specialCharRequired	Definiert, ob ein Sonderzeichen (*@#$%*) im Passwort enthalten sein soll **Mögliche Werte:** • *True*: Passwort wird ein Sonderzeichen enthalten • *False*: Passwort wird kein Sonderzeichen enthalten (Standard)

Statuswerte

`1 (successful)`	Die Operation wurde erfolgreich ausgeführt.
`-1 (failed)`	Die Operation konnte aufgrund eines technischen Fehlers nicht ausgeführt werden.

Input

Dieser Adapter erwartet keinen Input.

Ausgabe

Dieser Adapter gibt ein XML-Dokument mit folgender Struktur aus:

```
<Password><!-- Erzeugtes Passwort --></Password>
```

10.17.8 Random Number Generator

`Random Number Generator`: Erstellt Zufallszahlen basierend auf der angegebenen Konfiguration.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Generate`: generiert ein XML-Dokument mit einer oder mehreren Zufallszahlen • `SetRandomNumbers`: generiert Zufallszahlen in ein bereits vorhandenes XML-Dokument mittels XPath

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.randomNumberGenerator.RandomNumberGenerator*: Hauptklasse (Standard)
`min`	Minimum, ab welchem Zahlenwert eine Zufallszahlen generiert werden soll **Mögliche Werte:** beliebige positive Zahl, z. B. *1* Darf weder negativ noch `0` sein.
`max`	Maximum, ab welchem Zahlenwert eine Zufallszahlen generiert werden soll **Mögliche Werte:** beliebige positive Zahl, z. B. *100* Darf weder negativ noch `0` sein.
`count`	Anzahl der zu generierenden Zufallszahlen **Mögliche Werte:** beliebige positive Zahl, z. B. *4* Darf weder negativ noch `0` sein.
`XPath`	XPath zum gewünschten XML-Element Nur wichtig für die Operation `setRandomNumbers`. **Mögliche Werte:** Beliebiger XPath-Ausdruck, z. B. *Data/Beispiel*
`elementName`	Namen des XML-Elements in welchem die Zufallszahl(en) generiert werden soll(en). Nur wichtig für die Operation `setRandomNumbers`. **Mögliche Werte:** Beliebiger Name für ein XML-Element, z. B. *Id*

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt und im Output befindet sich ein XML-Dokument mit der/den Zufallszahl(en)
`-1`	Es kam zu einem Fehler während der Ausführung • Operationen `Generate` und `setRandomNumbers`: der Parameter `count` ist kleiner `1` • Operation `setRandomNumbers`: es liegt kein XML-Dokument als Input vor

Input

Der Adapter erwartet bei der Operation `SetRandomNumbers` ein XML-Dokument als Input.

Ausgabe

Je nach Operation, gibt der Adapter folgende Ergebnisse aus:

- Operation `Generate`: XML-Dokument mit der/den generierten Zufallszahl(en)
- Operation `SetRandomNumbers`: XML-Dokument mit der/den hinzugefügten Zufallszahl(en)

Beispiel

```
<?xml version="1.0" encoding="UTF-8">
<Data>
    <random>28</random>
    <random>6</random>
    <random>10</random>
</Data>
```

10.17.9 RSA Encryption

`RSA Encryption`: Verschlüsselt oder entschlüsselt Input-Daten (Byte-Datenstrom) mit dem RSA-Algorithmus und gibt einen Byte-Datenstrom aus (2048-Bit-RSA-Schlüssel).

Ehemalige Bezeichnung bis Version 4.5.1: *RSA Encryption Adapter*

> Die maximale Byte-Länge von verschlüsselten Input-Daten beträgt die Länge des RSA-Schlüssels in Bit dividiert durch 8, minus 11 Füllbytes. So berechnet sich beispielsweise die maximale Bytelänge bei einem 1024-Bit-RSA-Schlüssel wie folgt: 1024/8 -11 = 117 Bytes; Ein 2048-Bit-RSA-Schlüssel entschlüsselt Input-Daten mit maximal 245 Bytes Länge.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`Encrypt`*: Input-Daten RSA-verschlüsseln • *`Decrypt`*: RSA-verschlüsselte Input-Daten entschlüsseln

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.rsa.RSAAdapter*: Hauptklasse (Standard)
path	Pfad zur Keystore-Datei **Mögliche Werte:** Beliebiger absoluter oder relativer Pfad

alias	Kurzbezeichnung des Eintrags, der einen Schlüssel im Keystore besitzt **Mögliche Werte:** Beliebige Zeichenkette
keyType	Verwendeter Schlüsseltyp **Mögliche Werte:** • *private*: Privaten Schlüssel verwenden (Standard) • *public*: Öffentlichen Schlüssel verwenden
keystorePassword	Passwort für die Keystore-Datei **Mögliche Werte:** Beliebige Zeichenkette
privatePassword	Passwort, um den privaten Schlüssel aus der Keystore-Datei zu erhalten **Mögliche Werte:** Beliebige Zeichenkette

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.17.10 SAML Calculator

`SAML Calculator`: Signiert, validiert, verschlüsselt und entschlüsselt SAML Assertions und Requests.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • `Encrypt Assertion`: SAML Assertion verschlüsseln **Erforderliche Parameter:** `cryptAlias`, `cryptPassword` • `Decrypt Assertion`: SAML Assertion entschlüsseln **Erforderliche Parameter:** `cryptAlias`, `cryptPassword` • `Sign Assertion`: SAML Assertion mit einem Zertifikat signieren **Erforderliche Parameter:** `signatureAlias`, `signaturePassword`, `algorithm` • `Validate Assertion`: SAML Assertion validieren **Erforderliche Parameter:** `cryptAlias`, `cryptPassword`, `signatureRequired`, `skipValidation` • `Sign Request`: SAML Request mit einem Zertifikat signieren **Erforderliche Parameter:** `signatureAlias`, `signaturePassword`, `algorithm` • `Validate Request`: SAML Request validieren **Erforderliche Parameter:** `signatureRequired`, `skipValidations`

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.adapter.saml.SamlAdapter*: Hauptklasse (Standard)
`keystoreUrl`	URL des Keystores, der verwendet wird, um die SAML Assertions zu verschlüsseln und zu entschlüsseln. **Mögliche Werte:** String (URL)
`keystorePassword`	Passwort des Keystores, der verwendet wird, um die SAML Assertions zu verschlüsseln und zu entschlüsseln. **Mögliche Werte:** String
`keystoreType`	Art des Keystores, der verwendet wird, um die SAML Assertions zu verschlüsseln und zu entschlüsseln. **Mögliche Werte:** Art des Keystores, z. B. `JKS`, `PKCS12`, ...
`signatureRequired`	Legt fest, ob eine Signatur benötigt wird, um die Datei zu entschlüsseln. **Mögliche Werte:** `true` / `false`

`skipValidations`	Legt fest, ob die Prüfung von Signatur und Gültigkeit während des Entschlüsselns übersprungen werden sollen. **Mögliche Werte:** `true` / `false`
`cryptAlias`	Alias des Schlüsselpaars oder Zertifikats, das verwendet wird, um SAML Assertions zu verschlüsseln und zu entschlüsseln.
`cryptPassword`	Passwort des Schlüsselpaars, das verwendet wird, um SAML Assertions zu entschlüsseln.
`signatureAlias`	Alias des Schlüsselpaars, das verwendet wird, um SAML Assertions zu signieren.
`signaturePassword`	Passwort des Schlüsselpaars, das verwendet wird, um SAML Assertions zu signieren.
`algorithm`	Algorithmus, der verwendet werden soll, um SAML Assertions zu signieren. **Mögliche Werte:** • `SHA1`: • `SHA256`: • `SHA512`:

Statuswerte

1	Der Adapter gibt ein Ergebnis aus.
-1	Fehler bei der Ausführung des Adapters (Details siehe Server-Log)

Input

Der Adapter erwartet je nach gewählter Operation eine SAML Assertion oder ein SAML Request:

- SAML Assertion:

Beispiel Input SAML Assertion

```
 <saml:Assertion
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="_d71a3a8e9fcc45c9e9d248ef7049393fc8f04e5f75" Version="2.0"
    IssueInstant="2014-07-17T01:01:48Z">
    <saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
    <saml:Subject>
        <saml:NameID
            SPNameQualifier="http://sp.example.com/demo1/metadata.php"
            Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
            _ce3d2948b4cf20146dee0a0b3dd6f69b6cf86f62d7
        </saml:NameID>
        <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:
2.0:cm:bearer">
            <saml:SubjectConfirmationData NotOnOrAfter="2024-01-18T06:21:48Z"
                Recipient="http://sp.example.com/demo1/index.php?acs"
                InResponseTo="ONELOGIN_4fee3b046395c4e751011e97f8900b5273d56685
"/>
            </saml:SubjectConfirmation>
    </saml:Subject>
    <saml:Conditions NotBefore="2014-07-17T01:01:18Z" NotOnOrAfter="2024-01-18T
06:21:48Z">
        <saml:AudienceRestriction>
            <saml:Audience>http://sp.example.com/demo1/metadata.php</saml:Audie
nce>
        </saml:AudienceRestriction>
    </saml:Conditions>
    <saml:AuthnStatement AuthnInstant="2014-07-17T01:01:48Z"
        SessionNotOnOrAfter="2024-07-17T09:01:48Z"
        SessionIndex="_be9967abd904ddcae3c0eb4189adbe3f71e327cf93">
        <saml:AuthnContext>
            <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:
2.0:ac:classes:Password</saml:AuthnContextClassRef>
        </saml:AuthnContext>
    </saml:AuthnStatement>
    <saml:AttributeStatement>
        <saml:Attribute Name="uid"
            NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
            <saml:AttributeValue xsi:type="xs:string">test</saml:AttributeValue
>
        </saml:Attribute>
        <saml:Attribute Name="mail"
            NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
            <saml:AttributeValue xsi:type="xs:string">test@example.com</saml:At
tributeValue>
        </saml:Attribute>
        <saml:Attribute Name="eduPersonAffiliation"
            NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
```

```
            <saml:AttributeValue xsi:type="xs:string">users</saml:AttributeValu
e>
            <saml:AttributeValue xsi:type="xs:string">examplerole1</saml:Attrib
uteValue>
        </saml:Attribute>
    </saml:AttributeStatement>
</saml:Assertion>
```

- SAML Request:

Beispiel Input SAML Request

```
<samlp:AuthnRequest ID="123456789" Version="2.0"
    IssueInstant="2019-01-01T12:00:00"
    Destination="https://www.example.org/saml/login"
    ForceAuthn="false" IsPassive="false"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
    <saml:Issuer
        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">BPMX4</saml:Issuer>
    <samlp:NameIDPolicy AllowCreate="true"
        Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified" />
</samlp:AuthnRequest>
```

Ausgabe

Dieser Funktions-Adapter gibt je nach ausgeführter Operation eine andere Datei als Output aus:

- `Encrypt Assertion`: Der Adapter gibt eine entschlüsselte SAML-Assertion aus.
- `Sign Assertion`: Der Adapter gibt eine signierte SAML-Assertion aus.
- `Sign Request`: Der Adapter gibt einen signierte SAML-Request aus.

10.17.11 SSH Connector

`SSH Connector`: Verbindet sich mit einem SSH-Server über das Secure Shell Protocol.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Execute`: Führ den im Input eingegebenen Befehl auf dem Server aus

Parameter

`Adapter`	Adapter-Hauptklasse (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.ssh.SSHAdapter*: Hauptklasse (Standard)

`host`	Server-Host **Mögliche Werte:** Beliebiger Host oder IP-Adresse, z. B. *`localhost`* oder beliebige IP-Adresse
`user`	Benutzer, der sich auf dem Server verbinden soll **Mögliche Werte:** Beliebige Zeichenkette mit dem Benutzernamen, z. B. *`userName`*
`password`	Passwort zum Einloggen auf dem Server **Mögliche Werte:** Beliebige Zeichenkette mit dem Passwort, z. B. *`myPassword`*
`port`	Server-Port **Mögliche Werte:** • `22` (Standard) • beliebige ganze Zahl mit dem Port
`timeout`	Wartezeit, bis die Verbindung zum Server hergestellt ist in Millisekunden **Mögliche Werte:** • 30000 ms (Standard) • beliebige ganze Zahl für die Zeit in Millisekunden

Statuswerte

`0`	Die Adapter-Operation wurde erfolgreich ausgeführt
Beliebige Zahl außer `0`	Bei der Ausführung des Adapters kam es zu einem Fehler

Input

Der Adapter ein spezifisches XML-Dokument als Input:

Beispiel-Input

```
<?xml version="1.0" encoding="UTF-8"
<Root>
    <Command>netstat -plnt |grep 8080</Command>
</Root>
```

ⓘ Das Element `<Command>` enthält den auszuführenden Befehl.

Ausgabe

Der Adapter gibt ein XML-Dokument mit dem Output und dem Statuswert aus.

Ausgabe für das Beispiel-Input

```
<?xml version="1.0" encoding="UTF-8"
<Output>
    <CommandOutput>tcp 0 0 0.0.0.0:8080 0.0.0.0:* LISTEN - </CommandOutput>
    <ExitStatus>0</ExitStatus>
</Output>
```

10.17.12 TOTP One-Time Password Generator

`TOTP One-Time Password Generator`: Generiert ein TOTP-Einmalpasswort, das mit Google Authenticator und anderen Tools kompatibel ist.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `GenerateSecret`: 20 Byte lange Base32-Zeichenkette erstellen, die als geheimer Schlüssel für die Passwort-Generierung oder -Validierung oder als Basis für ein QR-Code-Bild verwendet werden kann • `GeneratePassword`: Einmalpasswort auf Basis des im XML-Inputdokument bereitgestellten geheimen Schlüssels erstellen • `VerifyPassword`: Einmalpasswort in Kombination mit dem geheimen Schlüssel überprüfen – beides wird im XML-Inputdokument angegeben • `GenerateAuthenticatorQRCode`: `512x512 px` Bilsd (`.png`) auf Basis des im XML-Inputdokument bereitgestellten geheimen Schlüssels erstellen ⓘ Diese Operation verwendet auch die Parameter `issuer` und `account`, um den QR-Code zu kommentieren. Diese Informationen werden sichtbar, wenn der Code von Google Authenticator oder einem ähnlichen Tool gescannt wird.

Parameters

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.adapter.totpgenerator.TOTPGenerator*: Hauptklasse (Standard)

`issuer`	Name der Firma oder Organisation, die das Einmalpasswort ausstellt ⓘ Dieser Parameter sollte wie ein Anzeigename behandelt werden, da er in der Authenticator App wie angegeben angezeigt wird, z. B. `SoftProject GmbH`.
`account`	Konto, das an den geheimen Schlüssel gebunden ist – typischerweise ein Benutzername oder eine E-Mail-Adresse ⓘ Dieser Parameter sollte wie ein Anzeigename behandelt werden, da er in der Authenticator App wie angegeben angezeigt wird, z. B. `John Doe` oder `john.doe@example.org`.

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Während der Ausführung der Operation kam es zu einem Fehler.
`403`	Das Passwort stimmt nicht überein, z. B. weil die Eingabestruktur nicht wohlgeformt ist.

Input

Der Adapter erwartet je nach gewählter Operation eine andere Inputstruktur.

- **Operation `GenerateSecret`:**
  Diese Operation benötigt keinen Input.
- **Operation `GeneratePassword`:**

  Input

  ```
  <Secret>
          <!-- 20-byte BASE32 character string -->
      </Secret>
  ```

- **Operation `VerifyPassword`:**

Input

```
<TOTP>
        <Password>
            <!-- 6-digit number with leading zeroes -->
        </Password>
        <Secret>
            <!-- 20-byte BASE32 character string -->
        </Secret>
    </TOTP>
```

- Operation **`GenerateAuthenticatorQRCode`**:

Input

```
<Secret>
    <!-- 20-byte BASE32 character string -->
</Secret>
```

Ausgabe

Der Adapter gibt abhängig von der gewählten Operation verschiedene Ausgabestrukturen aus.

- Operation **`GenerateSecret`**:

Output

```
<Secret>
    <!-- 20-byte BASE32 character string -->
</Secret>
```

- Operation **`GeneratePassword`**:

Output

```
<Password>
        <!-- 6-digit number with leading zeroes -->
    </Password>
```

- Operation **`VerifyPassword`**:

Output

```
<Ok description="The password matches."/>
            <!-- or -->
<Error description="The password provided is incorrect."/>
```

- **Operation `GenerateAuthenticatorQRCode`:**
  Ein Bild (`.png`) mit einem QR-Code wird ausgegeben.

10.17.13 UUID Generator

`UUID Generator`: Generiert universelle einzigartige Identifikatoren (UUIDs).

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • `Generate`: generiert ein XML-Dokument mit einer oder mehreren UUIDs • `SetUUID`: generiert UUIDs in ein bereits vorhandenes XML-Dokument mittels XPath-UUIDs

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.uuid.UuidGeneratorAdapter*: Hauptklasse (Standard)
`count`	Anzahl der zu generierenden UUIDs Nur wichtig für die Operation `Generate` **Mögliche Werte:** Beliebige positive Zahl, z. B. `1` Darf nicht negativ oder 0 sein.
`generateBraces`	Generierte UUIDs in eine geschweifte Klammer setzen **Mögliche Werte:** • `true`: generiert die UUID in einer geschweiften Klammer • `false`: generiert keine geschweifte Klammer
`xPath`	XPath zum gewünschten XML-Element Nur wichtig für die Operation `SetUUID` **Mögliche Werte:** Beliebiger XPath-Ausdruck, z. B. `Data/Beispiel`

`elementName`	Name des XML-Elements, in welchem die UUID generiert werden soll. Das Element kann bereits vorhanden sein und wird dann ggf. überschrieben. Nur wichtig für die Operation `SetUUID` **Mögliche Werte:** Beliebiger Name für ein XML-Element, z. B. `Id`

Statuswerte

`1`	Der Adapter wurde erfolgreich ausgeführt und in der Ausgabe befindet sich ein XML-Dokument mit der/den UUID(s)
`-1`	Der Adapter konnte nicht ausgeführt werden **Mögliche Gründe:** • Operation `Generate`: der Parameter `count` ist kleiner oder gleich `0` • Operation `SetUUID`: es liegt **kein** XML-Dokument als Input vor

Input

Der Adapter erwartet ein XML-Dokument für die Operation `Generate`.

Ausgabe

Das ausgegebene Dokument ist abhängig von der gewählten Operation:

- `Generate`: Der Adapter gibt ein XML-Dokument mit der/den generierten UUID(s) aus
- `SetUUID`: Der Adapter gibt ein XML-Dokument mit der/den hinzugefügten UUID(s) aus

Beispiel-Ausgabe

```
<?xml version="1.0" encoding="UTF-8"
<Data>
    <UUID>de41435e-f0a2-4fea-bdf7-9c8a261f8948</UUID>
    <UUID>fdeb6f6f-1abb-4d34-a405-a8f9a893e0ea</UUID>
    <UUID>d124a7fe-17d1-4850-a49c-6bdbe92a4c3c</UUID>
</Data>
```

10.17.14 XML Encryption

10.17.14.1 XML Encryption (Keystore)

? Unknown Attachment `XML Encryption (Keystore)`: Verschlüsselt oder entschlüsselt ein beliebiges XML-Dokument gemäß dem W3C-Standard XML-Encryption (XML-Enc) auf Basis eines Keystores vom Typ *PKCS12* oder *JCEKS*. Der Adapter unterstützt die Verschlüsselungsalgorithmen AES mit 128, 192 oder 256 Bit Schlüssellänge sowie Triple-DES. Für Ver-/Entschlüsselung auf Basis eines Passworts, siehe XML Encryption (Password).

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`Encrypt`*: XML-Dokument verschlüsseln • *`Decrypt`*: XML-Dokument entschlüsseln

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.xml.encryption.XmlEncryption*: Hauptklasse (Standard)
method	Algorithmus zur Verschlüsselung bzw. Entschlüsselung **Mögliche Werte:** • `AES_128`: AES-Algorithmus mit 128 Bit Schlüssellänge verwenden • `AES_192`: AES-Algorithmus mit 192 Bit Schlüssellänge verwenden • `AES_256`: AES-Algorithmus mit 256 Bit Schlüssellänge verwenden • `TRIPLEDES`: Triple-DES-Algorithmus verwenden
xpath	Pfad zum Element bzw. zu den Elementen, die verschlüsselt/entschlüsselt werden sollen als XPath-Ausdruck **Mögliche Werte:** Beliebiger gültiger XPath-Ausdruck, z.B. `//Data`
scope	Bestimmt den Wirkungsbereich für die Verschlüsselung innerhalb der per XPath adressierten XML-Dokumentstruktur **Mögliche Werte:** • `Element`: Das gesamte Element wird verschlüsselt • `Content`: Lediglich der Inhalt des Elements wird verschlüsselt
keyStoreURL	URL zu einem gültigen Keystore-Dokument **Mögliche Werte:** Beliebige gültige URL, z.B. `x4db:/1/Project/MyKeystore.pkcs12`

keyStoreType	Verwendeter Keystore-Typ **Mögliche Werte:** • `jceks`: Keystore im JCEKS-Format verwenden • `pkcs12`: Keystore im PKCS12-Format verwenden
password	Passwort zum Zugriff auf den Keystore **Mögliche Werte:** Beliebige Zeichenkette
alias	Alias für den Schlüssel im Keystore **Mögliche Werte:** Beliebige Zeichenkette
aliasPassword	Passwort um den Schlüssel über dessen o.g. Alias abzurufen (optional) **Mögliche Werte:** • Beliebige Zeichenkette • `(leer)`: Keystore-Passwort verwenden (Standard)

Statuswerte

-1 (error)	Fehler bei der Ausführung des Adapters (Details siehe Server-Log) Im Fehlerfall gibt der Adapter zudem eine Textdatei aus, welche die Exception enthält.
0 (empty)	Dokument wurde nicht verändert. Es wurden keine passenden Elemente gefunden.
1 (successful)	Dokument wurde verarbeitet und die entsprechenden Elemente ver-/entschlüsselt.

Beispiel

Im folgenden XML-Dokument sollen per XML-Encryption mit dem AES128-Algorithmus die Inhalte aller `Data`-Elemente verschlüsselt werden:

Input

```
<RootElement>
  <Data>Hello</Data>
  <Data>World!</Data>
  <Public>© SoftProject GmbH</Public>
</RootElement>
```

Der Adapter liefert dann beispielsweise folgendes verschlüsselte Ergebnis-XML-Dokument zurück:

Ergebnis

```
<RootElement>
   <Data>
      <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
         Type="http://www.w3.org/2001/04/xmlenc#Content">
         <xenc:EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
         <xenc:CipherData>
            <xenc:CipherValue>JS1lBp4RcMk6RepmptplJnBGqbpINGHSV9THG7+UJcY
            </xenc:CipherValue>
         </xenc:CipherData>
      </xenc:EncryptedData>
   </Data>
   <Data>
      <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
         Type="http://www.w3.org/2001/04/xmlenc#Content">
         <xenc:EncryptionMethod
          Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
         <xenc:CipherData>
            <xenc:CipherValue>rO+m1iPwsw4/PaP2AokM30DqsqeD1OYRvD7KNG+bnxc
            </xenc:CipherValue>
         </xenc:CipherData>
      </xenc:EncryptedData>
   </Data>
   <Public>© SoftProject GmbH</Public>
</RootElement>
```

10.17.14.2 XML Encryption (Password)

? Unknown Attachment `XML Encryption (Password):` Verschlüsselt oder entschlüsselt ein beliebiges XML-Dokument gemäß dem W3C-Standard XML-Encryption (XML-Enc) auf Basis eines Passworts. Der Adapter unterstützt die Verschlüsselungsalgorithmen AES mit 128, 192 oder 256 Bit Schlüssellänge sowie Triple-DES. Für Ver-/Entschlüsselung auf Basis eines Keystores, siehe XML Encryption (Keystore).

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *Encrypt*: XML-Dokument verschlüsseln • *Decrypt*: XML-Dokument entschlüsseln

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.xml.encryption.XmlEncryption*: Hauptklasse (Standard)
method	Algorithmus zur Verschlüsselung bzw. Entschlüsselung **Mögliche Werte:** • `AES_128`: AES-Algorithmus mit 128 Bit Schlüssellänge verwenden • `AES_192`: AES-Algorithmus mit 192 Bit Schlüssellänge verwenden • `AES_256`: AES-Algorithmus mit 256 Bit Schlüssellänge verwenden • `TRIPLEDES`: Triple-DES-Algorithmus verwenden
xpath	Pfad zum Element bzw. zu den Elementen, die verschlüsselt/ entschlüsselt werden sollen als XPath-Ausdruck **Mögliche Werte:** Beliebiger gültiger XPath-Ausdruck, z.B. `//Data`
scope	Bestimmt den Wirkungsbereich für die Verschlüsselung innerhalb der per XPath adressierten XML-Dokumentstruktur **Mögliche Werte:** • `Element`: Das gesamte Element wird verschlüsselt • `Content`: Lediglich der Inhalt des Elements wird verschlüsselt
password	Passwort, das zur Verschlüsselung und Entschlüsselung verwendet werden soll **Mögliche Werte:** Beliebige Zeichenkette

Statuswerte

-1 (error)	Fehler bei der Ausführung des Adapters (Details siehe Server-Log) Im Fehlerfall gibt der Adapter zudem eine Textdatei aus, welche die Exception enthält.

0 (empty)	Dokument wurde nicht verändert. Es wurden keine passenden Elemente gefunden.
1 (successful)	Dokument wurde verarbeitet und die entsprechenden Elemente ver-/entschlüsselt.

Beispiel

Im folgenden XML-Dokument sollen per XML-Encryption mit dem AES128-Algorithmus die Inhalte aller `Data`-Elemente verschlüsselt werden:

Input

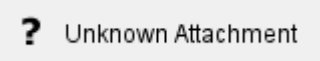
```
<RootElement>
  <Data>Hello</Data>
  <Data>World!</Data>
  <Public>© SoftProject GmbH</Public>
</RootElement>
```

Der Adapter liefert dann beispielsweise folgendes verschlüsselte Ergebnis-XML-Dokument zurück:

Ergebnis

```
<RootElement>
   <Data>
      <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
         Type="http://www.w3.org/2001/04/xmlenc#Content">
         <xenc:EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
         <xenc:CipherData>
            <xenc:CipherValue>JS1lBp4RcMk6RepmptplJnBGqbpINGHSV9THG7+UJcY
            </xenc:CipherValue>
         </xenc:CipherData>
      </xenc:EncryptedData>
   </Data>
   <Data>
      <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
         Type="http://www.w3.org/2001/04/xmlenc#Content">
         <xenc:EncryptionMethod
          Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
         <xenc:CipherData>
            <xenc:CipherValue>rO+m1iPwsw4/PaP2AokM30DqsqeD1OYRvD7KNG+bnxc
            </xenc:CipherValue>
         </xenc:CipherData>
      </xenc:EncryptedData>
   </Data>
   <Public>© SoftProject GmbH</Public>
</RootElement>
```

10.17.15 XML Signature



`XML Signature`: Fügt einem XML-Dokument eine Signatur hinzu oder überprüft eine Signatur innerhalb eines XML-Dokuments auf Basis einer Keystore-Datei im Format PKCS#12.
Ehemalige Bezeichnung bis Version 4.6: *Digital Signature Adapter*

Eigenschaften (Properties)

Custom Parameters	Benutzerdefinierte Parameter setzen; siehe Component-Parameter verwenden
Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • `Sign`: XML-Dokument signieren • `Check signature`: Gültigkeit der Signatur überprüfen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.digitalsignature.DigitalSignatureAdapter`*
keyStore	Pfad zu einer Keystore-Datei im Format PKCS#12 **Mögliche Werte:** • Relativer oder absoluter Pfad zu einem PKCS#12-Keystore • URL zu einem PKCS#12-Keystore • *Beispiel: `%USER_COLLECTION%/Project/Folder/Keystore.p12` oder `xstore://XMLSignatureXstore/Resources/KeyStore.p12`*
ksPass	Passwort des Keystores **Mögliche Werte:** Beliebige Zeichenkette
keyPass	Passwort für einen Schlüssel (optional) **Mögliche Werte:** Beliebige Zeichenkette

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Der Adapter erwartet für die Operation `Sign` ein beliebiges XML-Dokument als Input.

Output

Für die Operation `Sign` gibt der Adapter ein signiertes XML-Dokument aus.

10.17.16 AES Encryption

`AES Encryption` : Verschlüsselt und entschlüsselt Eingabedaten mit dem AES-Verschlüsselungsalgorithmus.

> **Hinweis:**
> **Advanced Encryption Standard (AES)**, ein öffentlich verfügbarer Verschlüsselungsalgorithmus, ist einer der sichersten und am häufigsten verwendeten Verschlüsselungsmethoden.
> Die Funktionsweise basiert auf einer Reihe von Byte-Ersetzungen, Verwürfelungen (Permutationen) und linearen Transformationen. Diese werden – daher der Begriff Blockverschlüsselung – auf 16-Byte-Datenblöcken durchgeführt. Die Operationen werden mehrmals wiederholt, und in jeder dieser Iterationen wird ein individueller, aus dem eigentlichen Schlüssel berechneter Iterationsschlüssel für die Berechnungen verwendet. AES als Verschlüsselungsmethode bietet einen großen Vorteil gegenüber klassischen Verschlüsselungsmethoden: Wenn nur ein einziges Bit im Schlüssel oder im Datenblock verändert wird, entsteht ein völlig anderer Chiffreblock. Die Länge des Schlüssels wird durch die Bezeichnungen `AES-128`, `AES-192` und `AES-256` spezifiziert: 128, 192 bzw. 256 Bit. AES ist der bevorzugte Verschlüsselungsstandard, denn bis heute ist für keine der AES-Varianten ein praktisch durchführbarer Angriff bekannt.

> Zusätzliche Informationen zur AES-Verschlüsselung finden Sie auch unter https://www.comparitech.com/blog/information-security/what-is-aes-encryption/.

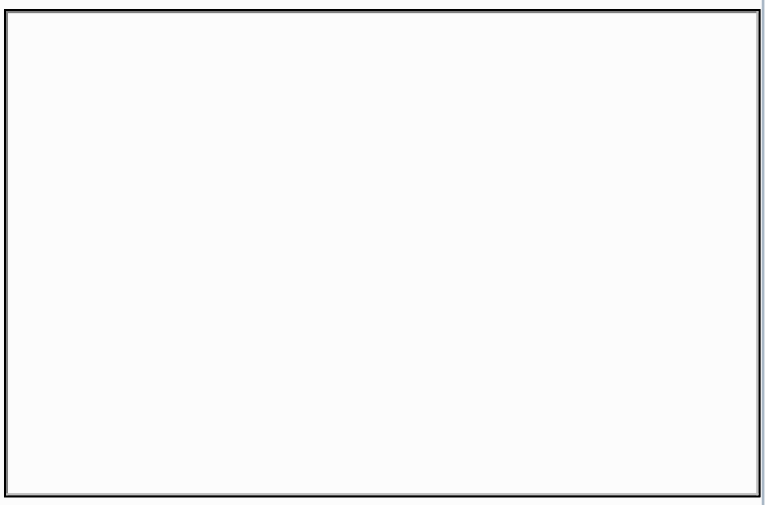
Eigenschaften (Properties)

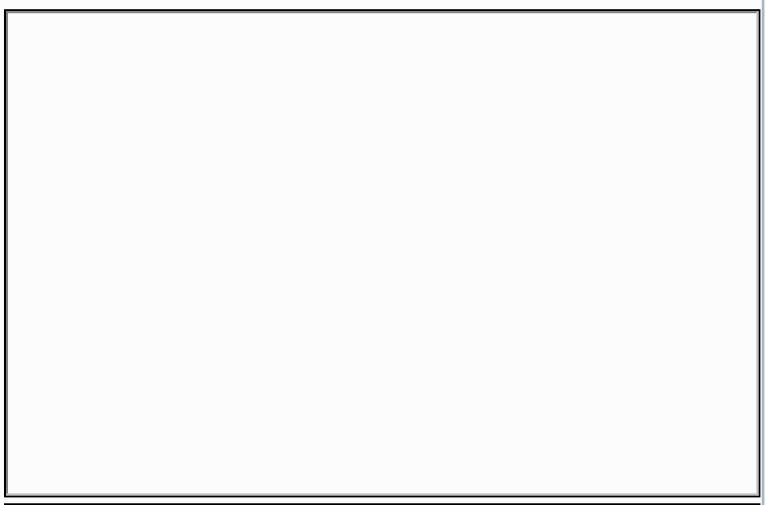
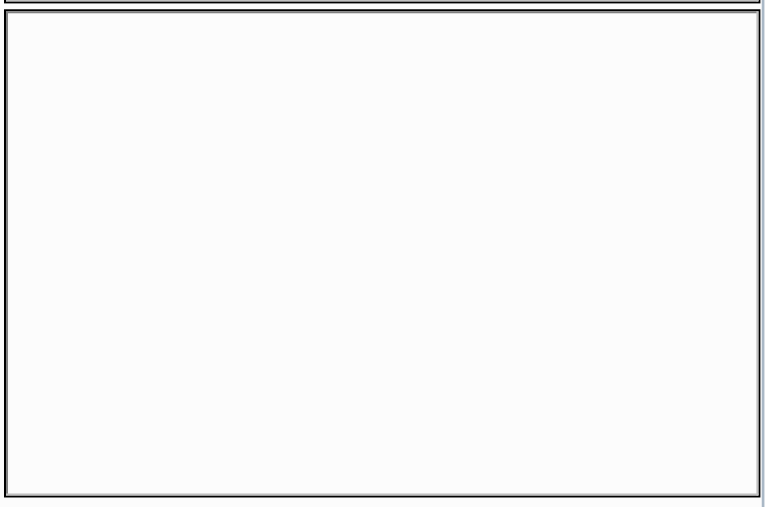
`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Encrypt`: Verschlüsselt Eingabedaten mit dem AES-Verschlüsselungsalgorithmus. • `Decrypt`: Entschlüsselt Eingabedaten mit dem AES-Verschlüsselungsalgorithmus. • `GenerateSharedKey`: Erzeugt einen Shared Key zum Verschlüsseln und Entschlüsseln der Eingabedaten. • `GenerateInitializationVector`: Erzeugt einen zufälligen Initialisierungsvektor zum Verschlüsseln und Entschlüsseln der Eingabedaten.

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern)! **Mögliche Werte:** *`de.softproject.integration.adapter.aes.AesEncryption`* : Hauptklasse (Standard)
`password`	*Erforderlich für die Operation `GenerateSharedKey`*. Passwort zum Erzeugen des Shared Keys.
`salt`	*Erforderlich für die Operation `GenerateSharedKey`*. Salt zum Erzeugen des Shared Keys (Base64-kodiert). In der Kryptografie bezeichnet Salt eine zufällig gewählte Zeichenfolge, die vor der weiteren Verarbeitung (z. B. Eingabe in eine Hash-Funktion) an einen bestimmten Klartext angehängt wird, um die Entropie der Eingabe zu erhöhen. Salt wird häufig zum Speichern und Übertragen von Passwörtern verwendet, um die Informationssicherheit zu erhöhen.

`keyAlgorithm`	*Erforderlich für die Operation `GenerateSharedKey`.* Algorithmus zum Erzeugen des Shared Keys. **Mögliche Werte:** • `DESede`: Erstellt Secret Keys zur Verwendung mit dem DESede- (Triple DES)-Algorithmus. • `PBEWith<digest>And<encryption>` bzw. `PBEWith<prf>And<encryption>` : Secret Key Factory zur Verwendung für PKCS5-Passwortverschlüsselung, wobei `<digest>` ein Nachrichtendigest, `<prf>` eine Pseudozufallsfunktion und `<encryption>` ein Verschlüsselungsalgorithmus ist, z. B. *`PBEWithMD5AndDES`* (PKCS #5, Version 1.5), *`PBEWithHmacSHA256AndAES_128`* (siehe PKCS #5, Version 2.0). • `PBKDF2With<prf>`: Passwort-basierter Algorithmus zur Schlüsselableitung (siehe PKCS #5, Version 2.0), der die Pseudozufallsfunktion `(<prf>)` nutzt, z. B. *`PBKDF2WithHmacSHA256`*
`keyLength`	*Erforderlich für die Operation `GenerateSharedKey`.* Schlüssellänge zum Erzeugen des Shared Keys. **Mögliche Werte:** • `128` • `196` • `256`
`keyIteration`	*Erforderlich für die Operation `GenerateSharedKey`.* Wiederholungen, die zum Erzeugen des Shared Keys verwendet werden soll.

`cipherAlgorihm`	Algorithmus zum Verschlüsseln und Entschlüsseln der Eingabedaten. **Mögliche Werte:** • `AES/CBC/PKCS5PADDING` • `AES/CBC/NOPADDING` • `AES/ECB/PKCS5PADDING` • `AES/ECB/NOPADDING` **Zusatzinformationen zum CBC- und zum ECB-Modus** • **Cipher Block Chaining Mode (CBC Mode)** ist eine Betriebsart, in der Blockchiffren betrieben werden können. Vor dem Verschlüsseln eines Klartextblocks wird dieser zunächst mit dem im vorhergehenden Schritt erzeugten Geheimtextblock per XOR (exklusives Oder) verknüpft. Die Struktur der Verschlüsselung und der Entschlüsselung im CBC-Modus ist in nachfolgenden Abbildungen dargestellt: 

	• **Electronic Code Book Mode (ECB Mode)** ist die einfachste Betriebsart für Blockchiffren, da jeder Klartextblock einzeln verschlüsselt wird. So ergeben gleiche Klartextblöcke unter Verwendung desselben Schlüssels immer denselben Chiffretextblock. Dies ist auch der große Nachteil dieser Methode, da die Klartextmuster erhalten bleiben. In nachfolgenden Abbildungen ist die Struktur des ECB bei der Ver- und Entschlüsselung dargestellt:  
`sharedKey`	Shared Key zum Verschlüsseln und Entschlüsseln der Eingabedaten (Base64-kodiert).

`initializationVector`	Initialisierungsvektor zum Verschlüsseln und Entschlüsseln der Eingabedaten; Funktioniert nur im CBC-Modus ⓘ Wenn Nachrichten verschlüsselt werden, muss vermieden werden, dass gleiche Klartextblöcke immer wieder gleiche Chiffretextblöcke ergeben. *Beispiel:* Im Deutschen beginnt ein förmlicher Brief in der Regel mit „Sehr geehrte(r) Frau/Herr“ gefolgt vom Namen. Mit diesem Wissen könnte ein Angreifer versuchen, Rückschlüsse auf den verwendeten Schlüssel zu ziehen. Um das zu vermeiden, wird der erste Klartextblock mit einem Initialisierungsvektor-XOR verknüpft. Da der Initialisierungsvektor zufällig erzeugt wurde, unterscheiden sich die entstehenden Chriffretexte auch dann, wenn die Klartexte mit identischen Daten beginnen.
`prependInitializationVector`	Legt fest, ob der Initialisierungsvektor als Präfix für die verschlüsselten Daten vorangestellt werden soll. ⓘ Da Angreifer ohne den Schlüssel keinen Informationsgewinn aus dem Initialisierungsvektor erhalten, kann der Initialisierungsvektor kann problemlos im Klartext an die verschlüsselten Daten übertragen werden. Der Initialisierungsvektor ist lediglich zur Verwässerung des Nachrichtenanfangs gedacht, bevor die Nachricht verschlüsselt wird. Wird der Initialisierungsvektor nicht übertragen, muss er auf anderem Wege erhalten werden, da er für die Entschlüsselung notwendig ist. **Mögliche Werte:** `true`- / `false`-Angabe
`encoding`	*Erforderlich für die Operation* `Decrypt`. Gibt das Encoding der entschlüsselten Daten an.
`mimeType`	*Erforderlich für die Operation* `Decrypt`. Gibt den Mime-Type der entschlüsselten Daten an.

Statuswerte

`-1`	Während der Ausführung der Operation kam es zu einem Fehler. Nähere Details finden Sie im Server-Log.
`1`	Die Operation wurde erfolgreich ausgeführt.

Input

Der Adapter erwartet je nach Operation verschiedene Input-Dokumente.

- Für die Operation`Encrypt` erwartet der Adapter ein beliebiges Input-Dokument, z. B. XML-, Text- oder Binär-Dateien.

Beispielhafter Input für Operation Encrypt

```
This is a top secret message and should be AES encrypted!
```

- Für die Operation `Decrypt` erwartet der Adapter AES-verschlüsselte Daten (Base64).

Beispielhafter Input für Operation Decrypt

```
dpi4dqEK7yiRcyz1HdLs9yxqNLIg6u8fqGR0JxADiCdHb/
6PGDqSxzID6VcuecxUP44x7bkHTggDpn8DO/bgKg==
```

Ausgabe

Der Adapter gibt je nach Operation verschiedene Ergebnis-Dokumente aus.

- Bei der Operation `Encrypt` gibt der Adapter die Eingabedaten verschlüsselt zurück (Base64-kodiert).

Beispielhafte Ausgabe für Operation Encrypt

```
dpi4dqEK7yiRcyz1HdLs9yxqNLIg6u8fqGR0JxADiCdHb/
6PGDqSxzID6VcuecxUP44x7bkHTggDpn8DO/bgKg==
```

- Bei der Operation `Decrypt` gibt der Adapter die verschlüsselten Eingabedaten entschlüsselt zurück.

Beispielhafte Ausgabe für Operation Decrypt

```
This is a top secret message and should be AES encrypted!
```

- Bei der Operation `GenerateSharedKey` gibt der Adapter den Shared Key mit den im Adapter definierten Key Parametern zurück.

Beispielhafte Ausgabe für Operation GenerateSharedKey

```
SfQYr8UlXmBIeHbZE28XkquKZHFX0qfsSTGBl+4WQDM=
```

- Bei der Operation `GenerateInitializationVector` gibt der Adapter einen zufälligen Initialisierungsvektor abhängig vom Cipher-Algorithmus aus.

Beispielhafte Ausgabe für Operation GenerateInitializationVector

```
JXAxGJ343MMjwLFi78z8xw==
```

Beispielszenario:

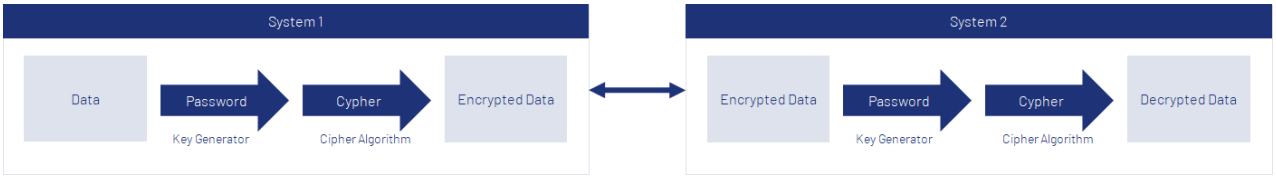
Ein typisches Szenario ist der Austausch von verschlüsselten Daten zwischen zwei Systemen. Eine Möglichkeit dies zu erreichen, ist ein geheimes Passwort zum Verschlüsseln und Entschlüsseln der Daten, das beide Systeme kennen. Dabei wird aus dem Passwort über den `GenerateSharedKey`-Algorithmus ein Schlüssel generiert. Der Schlüssel bereitet das Passwort kryptografisch auf und sorgt je nach gewähltem Algorithmus für eine stärkere Verschlüsselung. Je stärker der Schlüssel, desto schwerer können die verschlüsselten Daten durch Dritte entschlüsselt werden.

Alternativ zum Passwort kann auch der Schlüssel beiden Systemen bereitgestellt werden. Dies ist insofern weniger sicher, als das die zusätzliche Information, d.h. mit welchem Passwort der Schlüssel generiert wurde, nicht bekannt ist.

Mit dem Schlüssel und dem Cipher-Algorithmus werden die Daten für Dritte unkenntlich gemacht und können nur mit dem gleichen Schlüssel wieder kenntlich gemacht werden.

Die Verschlüsselten Daten können nun zum zweiten System transportiert werden, entweder über ein Netzwerk oder Hardware (USB-Stick). Auf diesem Weg kann kein Dritter die Daten verwenden.

Das zweite System erzeugt sich ebenfalls den Schlüssel mit den exakt gleichen Parametern wie das erste System. Zusammen mit dem wiederum gleichen Cipher-Algorithmus werden die Daten entschlüsselt und sind wieder verwendbar.



10.18 System Administration

- JMeter Test Execution
- JMX MBean
- JMX Number

- JMX Switch
- SNMP Management
- Windows Event Reader
- Windows Services Explorer

10.18.1 JMeter Test Execution

? Unknown Attachment `JMeter Test Execution`: Führt einen JMeter-Testplan aus und gibt ein Testergebnis als XML-Dokument zurück.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *`Execute`*: JMeter-Testplan ausführen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.jmeter.JMeterAdapter`*: Hauptklasse (Standard)
jmeterHome	Absoluter Pfad zum JMeter-Installationsordner **Mögliche Werte:**Beliebige Pfadangabe (z.B. *`c:\jakarta-jmeter-2.3.1\`*)
proxyHost	IP-Adresse oder Host-Name eines Proxy-Servers für JMeter **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *`192.168.1.1`*)
proxyPort	Portnummer des Proxy-Servers für JMeter **Mögliche Werte:** Beliebige ganze Zahl
proxyUser	Benutzername des Proxy-Servers für JMeter **Mögliche Werte:** Beliebige Zeichenkette
proxyPass	Passwort des Proxy-Servers für JMeter **Mögliche Werte:** Beliebige Zeichenkette

outputFormat	Ausgabeformat für das Testergebnis-Dokument **Mögliche Werte:** • *XML*: XML-Dokument ausgeben (Standard) • *CSV*: "Comma separated Value"-Textdatei ausgeben
timestamp	Zeitstempel-Angabe als Attribut ts im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ms*: Zeitstempel-Angabe in Millisekunden ausgeben (Standard) • *none*: Keinen Zeitstempel ausgeben
dataType	Datentyp als Attribut dt im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Datentyp ausgeben • *nein*: Keinen Datentyp ausgeben
label	Beschriftung (Label) als Attribut lb im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Beschriftung ausgeben • *nein*: Keine Beschriftung ausgeben (Standard)
responseCode	Antwortcode als Attribut rc im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Antwortcode ausgeben (Standard) • *nein*: Keinen Antwortcode ausgeben

responseData	Antwortdaten als Attribut responseData im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Antwortdaten ausgeben • *nein*: Keine Antwortdaten ausgeben (Standard)
responseDataOnError	Antwortdaten bei Fehlern als Attribut responseDataOnError im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Antwortdaten bei Fehlern ausgeben • *nein*: Keine Antwortdaten bei Fehlern ausgeben (Standard)
responseMessage	Antwortnachricht als Attribut rm im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Antwortnachricht ausgeben • *nein*: Keine Antwortnachricht ausgeben (Standard)
successful	Erfolgsmeldung als Attribut successful im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Erfolgsmeldung ausgeben (Standard) • *nein*: Keine Erfolgsmeldung ausgeben
time	Zeitangabe (in Millisekunden) als Attribut t im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Zeitangabe ausgeben (Standard) • *nein*: Keine Zeitangabe ausgeben

subresults	Unterergebnis als Attribut subresults im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Unterergebnis ausgeben • *nein*: Kein Unterergebnis ausgeben (Standard)
assertions	Aussagen als Attribut assertions im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Aussagen ausgeben • *nein*: Keine Aussagen ausgeben (Standard)
latency	Latenz als Attribut l im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Latenz ausgeben • *nein*: Keine Latenz ausgeben (Standard)
samplerData	Sampler-Daten als Attribut samplerData im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Sampler-Daten ausgeben • *nein*: Keine Sampler-Daten ausgeben (Standard)
responseHeaders	Antwort-Header als Attribut responseHeaders im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Antwort-Header ausgeben • *nein*: Keine Antwort-Header ausgeben (Standard)
requestHeaders	Request-Header als Attribut requestHeaders im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Request-Header ausgeben • *nein*: Keine Request-Header ausgeben (Standard)

encoding	Zeichenkodierung als Attribut c im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Zeichenkodierung ausgeben • *nein*: Keine Zeichenkodierung ausgeben (Standard)
bytes	Bytes als Attribut bytes im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Bytes ausgeben • *nein*: Keine Bytes ausgeben (Standard)
url	URL als Attribut url im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: URL ausgeben • *nein*: Keine URL ausgeben (Standard)
threadCounts	Anzahl der Threads als Attribut threadCounts im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Datentyp ausgeben • *nein*: Keinen Datentyp ausgeben (Standard)
filename	Dateiname als Attribut filename im Testergebnis-Dokument ausgeben **Mögliche Werte:** • *ja*: Dateiname ausgeben • *nein*: Keinen Dateiname ausgeben

assertionResults	Ergebnis als Feld assertionResults im Testergebnis-Dokument ausgeben (nur wenn Parameter outputFormat auf *CSV* gesetzt ist) **Mögliche Werte:** • *none*: Kein Ergebnis ausgeben (Standard) • *first*: Erstes Ergebnis ausgeben • *all*: Alle Ergebnisse ausgeben
printFieldNames	In der ersten Zeile des Testergebnis-Dokuments die Feldnamen definieren (nur wenn Parameter outputFormat auf *CSV* gesetzt ist) **Mögliche Werte:** • *ja*: Feldnamen in der ersten Zeile ausgeben • *nein*: Keine Feldnamen in der ersten Zeile ausgeben (Standard)
defaultDelimiter	Trennzeichen zwischen den einzelnen Datenfelder eines Datensatzes im Testergebnis-Dokument (nur wenn Parameter `outputFormat` auf *CSV* gesetzt ist) **Mögliche Werte:** • Beliebiges Zeichen außer Escape-Sequenzen wie *\t* • *,*: Komma (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.18.2 JMX MBean

? Unknown Attachment `JMX MBean`: Ermöglicht den Zugriff auf MBeans des lokalen JMX-Servers. Dabei können MBeans gesucht, MBean-Attribute gelesen und gesetzt, MBean-Operationen aufgerufen sowie MBean-Metadaten und Schlüsseleigenschaften separat ausgegeben werden. Dieser Adapter erwartet je nach gewählter Operation eine spezifische Input-XML-Struktur. Der Adapter gibt immer ein XML-Dokument aus, das in Abhängigkeit von der gewählten Operation Informationen über MBeans, die Ergebnisse der aufgerufenen MBean-Methoden oder ggf. Fehlermeldungen enthält.

Ehemalige Bezeichnung: *JMX MBean Manipulation Adapter*

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`List MBeans`*: Alle verfügbaren *MBeans* mit Name, Domäne und Schlüsseleigenschaften auflisten (kein Input-XML-Dokument erforderlich) • *`Find MBeans`*: MBean anhand des Objektnamens suchen; erwartet ein spezifisches Input-XML-Dokument; siehe Operation `Find MBeans` • *`Execute Operations`*: JMX-Operationen ausführen; erwartet je nach gewünschter Operation ein spezifisches Input-XML-Dokument

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.jmx.JmxAdapter`*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Adapter erwartet für die Operationen `Find MBeans` und `Execute Operations` eine adapterspezifische Input-XML-Struktur, siehe unten. Diese Input-XML-Struktur kann über XSL-Mappings in Abhängigkeit von den verarbeiteten Daten dynamisch generiert werden.

10.18.2.1 Operation: Find MBeans

Über die Operation `Find MBeans` können Sie nach MBeans suchen. Alle MBeans, die dem Suchmuster entsprechen, werden ausgegeben. Der Adapter erwartet bei dieser Operation folgendes Input-XML-Dokument.

```
<Elementname>
  <Query objectname="Suchausdruck" attribut="..." />
</Elementname>
```

Das Input-XML-Dokument besitzt ein beliebiges Wurzelement und beliebig viele Elemente `<Query>`. Jedes Element `<Query>` enthält im Attribut `objectname` einen Suchausdruck für den MBean-Objektnamen, der nach dem Muster `Domänenname:Schlüssel` bzw.

`Domänenname:Schlüsselname=Wert` aufgebaut ist; siehe http://java.sun.com/javase/6/docs/api/javax/management/ObjectName.html.

Der Suchausdruck kann die Platzhalter (*?* und ***) enthalten. Alle zusätzlich angegebenen Attribute werden unverändert zum Ausgabe-Dokument durchgereicht.

> **Beachten Sie:**
>
> - Je nach verwendetem Applikations-Server können Namensauflösungen fehlerhaft implementiert sein. So löst der MBean-Server von *JBoss* Platzhalter ggf. nicht auf, wenn Sie in Schlüsselnamen und -werten stehen.
> - Jeder Objektname muss durch einen Schlüssel qualifiziert werden. Wenn alle Instanzen des JMX-Objekts angesprochen werden sollen, ergänzen Sie den Objektnamen um *:**.

Beispiel

Die folgende Suchanfrage sucht innerhalb der Domänen `de.softproject.X0` bis `de.softproject.Xz` nach allen MBeans, deren Schlüssel (*Key Property*) `name` mit *X4* beginnt, und reicht ein Attribut `sampleAttribute="1"` ans Ausgabedokument weiter.

```
<JMX>
  <Query objectname="de.softproject.X?:name=X4*" sampleAttribute="1" />
</JMX>
```

Der Adapter gibt beispielsweise folgendes XML-Dokument mit den Sucherergebnissen aus:

```
<MBeans>
 <Query objectname="de.softproject.X?:name=X4*" sampleAttribute="value">
  <MBean objectname="de.softproject.X4:name=GlobalContextManager"
    domain="de.softproject.X4" keyProperties="name=GlobalContextManager">
    <KeyProperty name="name">GlobalContextManager</KeyProperty>
  </MBean>
  <MBean objectname="de.softproject.X4:name=X4Management"
    domain="de.softproject.X4" keyProperties="name=X4Management">
    <KeyProperty name="name">X4Management</KeyProperty>
  </MBean>
  </Query>
  </MBeans>
```

10.18.2.2 Operation: Execute Operations

Über die Operation `Execute Operations` können Sie MBean-Operationen aufrufen, Attribute lesen und setzen sowie MBean-Metadaten und Schlüsseleigenschaften separat ausgeben. Der grundsätzliche Aufbau des Input-XML-Dokument ist identisch wie bei der Operation `Find MBeans`, siehe Operation: Find MBeans, jedoch können innerhalb des Elements `<Query>` unterschiedliche

Elemente einfügt werden. Diese Elemente repräsentieren Anweisungen, die im Folgenden erläutert werden.

Beachten Sie:

- Wie bei Operation `Find MBeans` kann auch bei Operation `Execute Operations` der Suchausdruck in Attribut `objectname` die Platzhalter *?* und *** enthalten.
- Jeder Objektname muss durch einen Schlüssel qualifiziert werden. Wenn alle Instanzen des JMX-Objekts angesprochen werden sollen, ergänzen Sie den Objektnamen um *:**.

Metadaten auslesen

Das Kindelement `<GetMetadata/>` des Elements `<Query>` weist den Adapter an, sämtliche Metadaten der entsprechenden MBeans auszugeben:

- Im Attribut `class`: Nennt die Implementierungsklasse
- `<Description>`: Enthält einen Beschreibungstext der MBean
- `<Constructors>`: Listet Konstruktoren mit zugehörigen Operationen und Beschreibung auf
- `<Attributes>`: Führt alle Attribute der MBean auf
- `<Operations>`: Führt alle Operationen der MBean mit Name, Beschreibung, Datentypen, Parametern usw.
- `<Notifications>`: Führt ggf. Benachrichtigungen der MBean auf

```
<Query objectname="Suchausdruck" attribut="wert">
   <GetMetadata/>
</Query>
```

Schlüsseleigenschaften separat ausgeben

Das Kindelement `<ExpandKeyProperties/>` des Elements `<Query>` weist den Adapter an, alle Schlüsseleigenschaften in jeweils einem separaten Element `<Property>` auszugeben:

```
<Query objectname="Suchausdruck" attribut="wert">
   <ExpandKeyProperties/>
</Query>
```

Attributwerte auslesen

Das Kindelement `<GetAttributeValue/>` des Elements `<Query>` weist den Adapter an, Werte zu MBean-Attributen auszugeben, die in jeweils einem Kindelement `<Attribute>` im Attribut `name` angegeben werden:

```
<Query objectname="Suchausdruck" attribut="wert">
   <GetAttributeValue>
      <Attribute name="Attribut-Name" />
   </GetAttributeValue>
</Query>
```

Attributwerte auslesen

Das Kindelement `<SetAttributeValue/>` des Elements `<Query>` weist den Adapter an, MBean-Attributwerte zu setzen. Dabei wird für jedes MBean-Attribut in einem Element `<Attribute>` festgelegt:

- Im Attribut `name`: Name des zu setzenden MBean-Attributs
- Im Attribut `class`: Datentyp des zu setzenden MBean-Attributs
- Der Wert des MBean-Attributs wird im Elementinhalt angegeben.

```
<Query objectname="Suchausdruck" attribut="wert">
   <SetAttributeValue>
      <Attribute name="Attribut-Name" class="Datentyp"></Attribute>
   </SetAttributeValue>
</Query>
```

MBean-Operationen aufrufen

Das Kindelement `<InvokeOperation/>` des Elements `<Query>` weist den Adapter an, eine MBean-Operation aufzurufen. Das Element `<InvokeOperation/>` muss im Attribut `name` den Namen der MBean-Operation enthalten. In beliebig vielen Kindelementen `<Parameter>` wird der jeweilige Parameterwert angegeben, im Attribut `class` wird der Datentyp des Parameterwerts definiert.

Folgende Datentyp-Angaben im Attribut class können verwendet werden:

• *boolean*	• *double*	• *java.lang.Float*
• *char*	• *java.lang.Boolean*	• *java.lang.Double*
• *byte*	• *java.lang.Character*	• *java.lang.String*
• *short*	• *java.lang.Byte*	• *java.math.BigInteger*
• *int*	• *java.lang.Short*	• *java.math.BigDecimal*
• *long*	• *java.lang.Integer*	• *java.management.ObjectName*
• *float*	• *java.lang.Long*	

ⓘ Da MBean-Parameter keine Namen besitzen, ist die Reihenfolge maßgeblich, in der diese notiert werden.

Ein Aufruf einer MBean-Operation erfolgt also durch eine nach folgendem Muster aufgebaute Input-XML-Struktur:

```
<Query objectname="Suchausdruck" attribut="wert">
   <InvokeOperation name="MBean-Operation">
      <Parameter class="Datentyp">Parameterwert</Parameter>
   </InvokeOperation>
</Query>
```

Beispiel

Im folgenden Input-XML-Dokument werden Metadaten abgefragt, Schlüsseleigenschaften gelistet, MBean-Attribute gelesen und gesetzt sowie eine MBean-Operation mit zwei Parametern ausgeführt:

```
<RootElement>
   <Query objectname="de.softproject.X?:name=X4Management">
      <GetMetadata/>
   </Query>
   <Query objectname="java.lang*:*">
      <ExpandKeyProperties/>
   </Query>
   <Query objectname="de.softproject.X4.WS*:*" text="enabling WebServices">
      <GetAttributeValue>
         <Attribute name="InService"/>
      </GetAttributeValue>
      <SetAttributeValue>
         <Attribute name="InService" class="boolean">true</Attribute>
      </SetAttributeValue>
   </Query>
   <Query objectname="de.softproject.X4:name=X4Management"
      description="Start a process">
      <InvokeOperation name="startProcessByName">
         <Parameter class="java.lang.String">1</Parameter>
         <Parameter class="java.lang.String">/Project/Folder/Process.wrf
         </Parameter>
      </InvokeOperation>
   </Query>
   </RootElement>
```

10.18.3 JMX Number

`JMX Number`: Ermöglicht die Überwachung des aktuell ausgeführtenTechnical Processes in Form eines Prüfpunkts, der jeweils einer JMX-MBean entspricht.
Wenn der Prüfpunkt erfolgreich durchlaufen wird, dann wird der Zeitstempel der MBean aktualisiert. Der zurückgegebene Adapter-Status hängt vom Wert der MBean-Eigenschaft `Value` ab, der einen ganzzahligen Wert zwischen *0* und *899* annehmen kann. Über den jeweiligen nummerischen Adapter-Status können mit diesem Adapter auch komplexere Fallunterscheidungen im Technical Process abgebildet werden.

Falls die entsprechende MBean noch nicht existiert, wird diese angelegt, mit einem aktuellen Zeitstempel versehen und die MBean-Eigenschaft `Value` auf den Parameterwert von `Startvalue`

gesetzt. Zudem kann über den Adapter eine bestehende Prüfpunkt-MBean aus dem JMX-Register wieder entfernt werden.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`Check`*: Zeitstempel aktualisieren und MBean-Wert `value` einlesen • *`Unregister`*: MBean aus dem JMX-Register entfernen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.jmxtools.JmxNumberAdapter`*: Hauptklasse (Standard)
Objectname	Name des bearbeiteten Objekts (Prüfpunkt) **Mögliche Werte:** • *`de.softproject.x4.jmxnumber:pid=%PID%,process=%CURRENT_PROCESS%,userid=%USER_ID%`*: aktuelle Prozess-Instanz (Standard) • Beliebiger gültiger JMX-Objektname, siehe http://java.sun.com/javase/6/docs/api/javax/management/ObjectName.html)
Startvalue	Standardwert für die MBean-Eigenschaft `Value` **Mögliche Werte:** • Beliebige ganze Zahl zwischen *`0`* und *`899`* • *`1`*: Wert *`1`* zur Ausgabe von Status *`1`* (Standard)
Persistent	Wert der JMX-Bean persistieren **Mögliche Werte:** • *`Ja`*: Wert persistieren • *`Nein`*: Nicht persistieren (Standard)

Input

Der Funktions-Adapter nimmt beliebige Input-Dokumente entgegen und gibt diese unverändert aus.

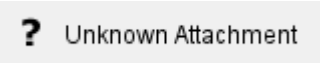
Ausgabe

Das ausgegebene Dokument entspricht dem Input-Dokument.

Statuswerte

0...899	MBean-Eigenschaft Value besitzt einen Wert zwischen *0* und *899*
-1	Fehler beim JMX-Aufruf (z. B. inkompatible MBean)

10.18.4 JMX Switch



`JMX Switch`: Ermöglicht die Überwachung des aktuell ausgeführten *X4*-Prozesses in Form eines Prüfpunkts, der jeweils einer JMX-MBean entspricht.

Wenn der Prüfpunkt erfolgreich durchlaufen wird, dann wird der Zeitstempel der MBean aktualisiert. Der zurückgegebene Adapter-Status hängt vom Wert der MBean-Eigenschaft `Value` ab. Falls die entsprechende MBean noch nicht existiert, wird diese angelegt, mit einem aktuellen Zeitstempel versehen und die MBean-Eigenschaft `Value` auf den Parameterwert von `Startvalue` gesetzt. Zudem kann über den Adapter eine bestehende Prüfpunkt-MBean aus dem JMX-Register wieder entfernt werden.

Ehemalige Bezeichnung bis Version 4.5.1: *JMX Switch Adapter*

> ⓘ Mit dem Adapter *JMX Number* lässt sich statt eines Booleschen ein ganzzahliger Wert von *0* bis einschließlich *899* setzen. Über den jeweiligen nummerischen Adapter-Status können mit diesem Adapter, auch komplexere Fallunterscheidungen im *X4*-Prozess abgebildet werden, siehe JMX Number.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`Check`*: Zeitstempel aktualisieren und MBean-Wert `value` einlesen • *`Unregister`*: MBean aus dem JMX-Register entfernen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.jmxtools.JmxSwitchAdapter*: Hauptklasse (Standard)

Objectname	Name des bearbeiteten Objekts (Prüfpunkt) **Mögliche Werte:** • *de.softproject.x4.jmxswitch: pid=%PID%,process=%CURRENT_PROCESS%,userid=%USER_ID%*: aktuelle Prozess-Instanz (Standard) • Beliebiger gültiger JMX-Objektname; siehe http://java.sun.com/javase/6/docs/api/javax/management/ObjectName.html
Startvalue	Standardwert für die MBean-Eigenschaft Value **Mögliche Werte:** • `Ja`: True (Standard) • `Nein`: False
Persistent	Wert der JMX-Bean persistieren **Mögliche Werte:** • `Ja`: Wert persistieren • `Nein`: Nicht persistieren (Standard)

Input

Der Funktions-Adapter nimmt beliebige Input-Dokumente entgegen und gibt diese unverändert aus.

Ausgabe

Das ausgegebene Dokument entspricht dem Input-Dokument.

Statuswerte

0	MBean-Eigenschaft Value besitzt den Wert *false*
1	MBean-Eigenschaft Value besitzt den Wert *true*
-1	Fehler beim JMX-Aufruf (z. B. inkompatible MBean)

10.18.5 SNMP Management

`SNMP Management`: Verwaltet Geräte in einem IP-Netzwerk über das *Simple Network Management Protocol* (SNMP) in Version 1, 2c oder 3 (mit automatisch gewählter Sicherheitsebene). Der Adapter fordert einzelne oder mehrere Management-Datensätze (MIB-Objekte) von Agenten an, setzt Werte von Management-Datensätzen und versendet Trap-Nachrichten an Agenten.

Dieser Adapter erwartet ein Input-XML-Dokument mit einer spezifischen Struktur, die MIB-Objekt-IDs und SNMP-Parameter enthält, und gibt ein XML-Dokument aus.

> ⓘ Wenn der Agent bei der Überwachung des Systems einen Fehler erkennt, kann er diesen mit Hilfe eines Trap-Paketes unaufgefordert an den *SNMP Management Adapter* melden. Diese Pakete werden nicht vom Adapter bestätigt. Der Agent kann damit nicht feststellen, ob der Trap beim Adapter angekommen ist.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *GET*: Einzelnen Management-Datensatz (MIB-Objekt) anfordern. Der Agent antwortet mit einem Response-Paket, das entweder die angeforderten Daten oder eine Fehlermeldung enthält. • *GETNEXT*: Nachfolgenden Management-Datensatz (MIB-Objekt) anfordern (um Tabellen zu durchlaufen) • *GETBULK*: Mehrere Management-Datensätze (MIB-Objekte) gleichzeitig anfordern • *SET*: Einen oder mehrere Management-Datensätze (MIB-Objekte) eines Geräts verändern oder Aktionen auszulösen. Der Agent bestätigt die Übernahme der Werte mit einem Response-Paket. • *SENDTRAP*: Trap-Nachricht zu einem Ereignis unaufgefordert an den Manager per UDP von Port 161 senden

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.snmp.SnmpAdapter*: Hauptklasse (Standard)
host	IP-Adresse oder Host-Name der Netzwerk-Management-Station (NMS) **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *127.0.0.1*)

port	UDP-Portnummer (nur für SNMP in Version 2c und 3 änderbar) **Mögliche Werte:** • *161*: Standard-Port, auf dem SNMP-Agenten horchen für die Operationen *GET*, *GETNEXT*, *GETBULK* und *SET* (Standard) • *162*: Port, auf dem die Netzwerk-Management-Station standardmäßig horcht (für die Operation *SENDTRAP*)
version	Verwendete SNMP-Version **Mögliche Werte:** • *version1*: Version 1 verwenden • *version2c*: Version 2c verwenden • *version3*: Version 3 verwenden
timeout	Zeitbeschränkung für den Verbindungsaufbau in Millisekunden **Mögliche Werte:** • Beliebige ganze Zahl • *3*: Zeitbeschränkung von 3 ms (Standard)
retries	Anzahl der Wiederholung von Verbindungsversuchen, falls die Verbindung nicht direkt aufgebaut werden kann **Mögliche Werte:** • Beliebige ganze Zahl • *0*: Kein zusätzlicher Verbindungsversuch (Standard)
walk	Sämtliche Anfragen im Input-XML-Dokument mit der *Walk* -Operation ausführen, bis alle Instanzen im OID-Teilbaum abgefragt wurden (wenn die Eigenschaft Operation auf *GETNEXT* oder *GETBULK* gesetzt ist) **Mögliche Werte:** • *true*: *GETNEXT*- und *GETBULK*-Anfragen mit der *SNMP Walk*-Operation ausführen • *false*: Keine *Walk*-Operation ausführen, wenn nicht anders im Input-XML-Dokument definiert (Standard)

contextEngineId	SNMP-Entity, die die Anfrage ausführen soll (wenn Parameter `version` auf *`version3`* gesetzt ist) **Mögliche Werte:** Beliebige Zeichenkette
community	Name der Community der SNMP-Agenten (wenn der Parameter version auf *`version1`* oder *`version2c`* gesetzt ist) **Mögliche Werte:** Beliebige Zeichenkette (z. B. *`public`*)
securityname	Sicherheitsnamen zur Authentifizierung von Nachrichten über SNMP Version 3 (wenn Parameter `version` auf *`version3`* gesetzt ist) **Mögliche Werte:** Beliebige Zeichenkette
authenticationprotocol	Verwendetes Verfahren zur Authentifizierung einer Nachricht über SNMP Version 3 (wenn Parameter `version` auf *`version3`* gesetzt ist) **Mögliche Werte:** • *`MD5`*: *MD5*-Prüfsumme erstellen • *`SHA`*: *Secure Hash*-Code erstellen
authenticationpass	Passwort zur Authentifizierung über SNMP Version 3 auf Sicherheitsebene *authNoPriv* (wenn Parameter `version` auf *`version3`* gesetzt ist) **Mögliche Werte:** Beliebige Zeichenkette mit mindestens 8 Zeichen Länge
privacyprotocol	Verschlüsselungs-Algorithmus für Nachrichten mit SNMP Version 3 (wenn Parameter `version` auf *`version3`* gesetzt ist) **Mögliche Werte:** • *`DES`*: *Data Encryption Standard*-Algorithmus verwenden • *`AES`*: *Advanced Encryption Standard*-Algorithmus verwenden • *`AES192`*: *Advanced Encryption Standard*-Algorithmus mit 192 Bit Schlüssellänge verwenden • *`AES256`*: *Advanced Encryption Standard*-Algorithmus 256 Bit Schlüssellänge verwenden

privacypass	Passwort für die Verschlüsselung von Nachrichten mit SNMP Version 3 auf Sicherheitsebene `authPriv` (wenn Parameter `version` auf *`version3`* gesetzt ist) **Mögliche Werte:** Beliebige Zeichenkette

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Adapter erwartet eine adapterspezifische Input-XML-Struktur, wobei jedes Element `<pdu/>` für ein Netzwerkgerät steht.

```
<root>
    <pdu oid="MIB-Objekt-ID" timeout="Zeitbeschränkung in ms"
         retries="Anzahl zusätzlicher Verbindungsversuche" />
</root>
```

Ausgabe

Dieser Adapter gibt ein UTF-8-kodiertes XML-Dokument mit folgender Struktur aus.

```
<root>
    <pdu oid="MIB-Objekt-ID">
        <![CDATA[ Ausgegeben Daten des MIB-Objekts ]]>
    </pdu>
</root>
```

10.18.6 Windows Event Reader

? Unknown Attachment `Windows Event Reader`: Liest das Event Log (Ereignisprotokoll) von Microsoft Windows und gibt sämtliche oder nach Filterkriterien gewählte Ereignis-Einträge in einem XML-Dokument aus.
Ehemalige Bezeichnung: *Windows Event Logging Adapter*

ⓘ Folgende Bedingungen müssen erfüllt sein, um diesen Adapter verwenden zu können:

- Der Windows-Computer, von dem das Ereignisprotokoll gelesen werden soll, muss in derselben Domäne liegen wie der *X4 Server.*
- Der *X4 Server* muss auf einem Windows-Computer laufen, der Domänen-Administratoren-Rechte besitzt.
- Eine JNI-Lizenz wird benötigt.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *Execute*: Adapter ausführen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.windowseventlog.WindowsEventLogAdapter*: Hauptklasse (Standard)
log	Windows-Protokoll/Log, dessen Ereignisse angezeigt werden sollen **Mögliche Werte:** • Beliebiges Windows-Protokoll/Log (z.B. *Application*) • *System*: Systemereignisse ausgeben (Standard)
from	Startdatum zur Beschränkung auf einen Zeitraum **Mögliche Werte:** Beliebiges Datum und Uhrzeit im Format `YYYY-MM-DDThh:mm:ss` (z.B. *2012-03-31T13:45:03*)
to	Enddatum zur Beschränkung auf einen Zeitraum **Mögliche Werte:** Beliebiges Datum und Uhrzeit im Format YYYY-MM-DDThh:mm:ss (z.B. *2012-03-31T13:45:03*)
type	Nach einem Ereignis-Typ (Ebene) filtern **Mögliche Werte:** Beliebiger Ereignis-Typ (z.B. *Error)*
source	Nach einer Ereignis-Quelle (dies kann beispielsweise eine Anwendung sein) filtern **Mögliche Werte:** Beliebige Zeichenkette (z.B. *PrintSpooler*)
category	Nach einer Ereignis-Kategorie filtern **Mögliche Werte:** Beliebige Zeichenkette
eventId	Nach einer Ereignis-ID filtern **Mögliche Werte:** Beliebige ganze Zahl

user	Nach einem Benutzer filtern **Mögliche Werte:** Beliebige Zeichenkette (z.B. *SYSTEM*)
computer	Name des Computers, von dem das Ereignisprotokoll gelesen werden soll **Mögliche Werte:** • Beliebiger Computer-Name oder IP-Adresse • *127.0.0.1*: Momentan genutztes System, d.h. localhost (Standard)

Statuswerte

1	Die Operation wurde erfolgreich ausgeführt.
-1	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Ausgabe

Dieser Funktions-Adapter gibt ein UTF-8-kodiertes XML-Dokument mit folgender Struktur aus. Für jedes Ereignis wird ein XML-Element `<log/>` erzeugt, das eine Reihe von Informationen sowie die Ereignis-Beschreibung zum Ereignis enthält:

```
<logs>
    <log type="Ereignistyp" date="Datum und Uhrzeit"
        source="Quelle" category="Kategorie" event="Ereignis-ID" user="Benutzer"
        computer="Computername">Ereignis-Beschreibung</log>
</logs>
```

10.18.7 Windows Services Explorer

`Windows Services Explorer`: Erzeugt eine Liste von Windows-Diensten und deren Status und gibt diese in einem XML-Dokument aus.

Folgende Voraussetzungen müssen erfüllt sein, um diesen Adapter verwenden zu können:

- Der Windows-Computer, von dem Dienst-Statusinformationen gelesen werden sollen, muss in derselben Domäne liegen wie der *X4 Server.*
- Der *X4 Server* muss auf einem Windows-Computer laufen, der Domänen-Administratoren-Rechte besitzt.
- Eine JNI-Lizenz wird benötigt.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *Execute*: Adapter ausführen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.windowsservice.WindowsServiceAdapter*: Hauptklasse (Standard)
host	Name des Computers, der den Windows-Dienst ausführt **Mögliche Werte:** • Beliebiger Computer-Name oder IP-Adresse • *127.0.0.1*: Momentan genutztes System, d.h. localhost (Standard)
service	Nach einem bestimmten Windows-Dienst filtern **Mögliche Werte:** Beliebige Zeichenkette

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Ausgabe

Dieser Adapter gibt ein UTF-8-kodiertes XML-Dokument mit folgender Struktur aus. Für jeden Dienst wird ein XML-Element `<service/>` erzeugt, das Informationen zum Windows-Dienst sowie dessen Status enthält:

```
<services>
   <service name="Dienst-Name" dispalyName="Anzeigename des Diensts"
            startupType="Starttyp" logOnAs="Benutzer/Anmeldetyp">
            <!--Status des Dienstes--></service>
</services>
```

10.19 Tools

- WebSMS

10.19.1 WebSMS

`WebSMS`: Stellt eine Verbindung zu websms her und versendet SMS.

Um sich mit websms zu verbinden und SMS zu versenden, wird ein websms Benutzerkonto benötigt. Zusätzlich muss die Zwei-Faktor-Authentifizierung aktiviert sein.

> ⓘ Weitere Informationen finden Sie auf der websms Webseite.

Parameter

`Adapter`	Adapter Hauptklasse (nicht ändern!) **Mögliche Werte:** • *`de.softproject.x4.adapter.checkrights.CheckRightsAdapter`*: Hauptklasse (Standard)
`targetPhoneNumbers`	Telefonnummern der Empfänger im MSISDN-Format. Mehrere Telefonnummern werden durch ein Komma getrennt.
`websmsUsername`	Benutzername des websms Benutzerkontos
`websmsPassword`	Passwort des websms Benutzerkontos
`websmsAccessToken`	API-Token ⚠ Der API-Token muss in der API-Verwaltung von websms erstellt werden.
`deliveryReportCallbackURL`	URL für das Weiterleiten von Zustellungsberichten. ⓘ (websms API: notificationCallbackUrl)
`simulation`	Simulationsmodus: Wenn der Simulationsmodus aktiviert ist, wird keine SMS versendet. ⓘ (websms API: test)

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters ist ein Fehler aufgetreten.

Input

Der Adapter erwartet ein XML-Dokument als Input. Nur der Inhalt des Wurzelelements wird als SMS versendet.

```
<root>
  This text will be sent as SMS.
<root>
```

10.20 Web Services

- JSON
- OData Connector
- URL Encoding Converter
- URL Reader
- Webservice
- X4 Streaming Support
- XOP (MTOM)
- YAML Converter

10.20.1 JSON

10.20.1.1 JSON Converter 2.0

`JSON Converter 2.0`: Konvertiert ein Textdokument, das strukturierte Daten in der *JavaScript Object Notation (JSON)* enthält, in ein XML-Dokument, und konvertiert umgekehrt ein XML-Dokument in ein UTF-8-kodiertes JSON-Dokument. Dabei verwendet der Adapter ein neues XML-Format, das eine bessere Kontrolle des generierten JSON-Dokuments erlaubt.

Dieser Adapter ersetzt den veralteten JSON Converter.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • `JSON2XML`: JSON-Dokument in XML konvertieren • `XML2JSON`: XML-Dokument in JSON konvertieren

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.json.Json2Converter`*: Hauptklasse (Standard)

`allowBareObjectKeys`	Angaben zum Umgang mit Object Keys, die nicht in Anführungszeichen stehen **Mögliche Werte:** • `true`: erlaubt Object Keys, die nicht in Anführungszeichen stehen • `false`: erlaubt keine Object Keys, die nicht in Anführungszeichen stehen
`escapeStringInXml`	Angaben zum Umgang mit Nicht-XML-Zeichen **Mögliche Werte:** • `true`: Nicht-XML-Zeichen escapen. Zeichen, die in XML verboten sind, werden durch `\u????` ersetzt, wobei die Fragezeichen den Hexadezimalen Ziffern des Unicode-Zeichens entsprechen • `false`: Nicht-XML-Zeichen filtern. Zeichen, die in XML verboten sind werden durch das Unicode-Zeichen `uFFFD` (Replacement Character) ersetzt
`emptyStringAsNull`	Leere Zeichenketten bei Zahlen und Boolean-Werten als `null` ausgeben **Mögliche Werte:** • `true`: Leere Zeichenketten als `null` ausgeben, z. B. *`<Value type="Boolean"></Value>`* wird als null ausgegeben • `false`: Leere Zeichenketten nicht als `null` ausgeben, z. B. *`<Value type="Boolean"></Value>`* erzeugt einen Fehler ⓘ `<Value type="String"></Value>` erzeugt immer `""`.

ⓘ Bei der Operation `XML2JSON` werden für nummerische und boolesche Werte Whitespaces am Anfang und Ende des Wertes ignoriert.

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Beispiel

Folgende Beispiele zeigen JSON-Ausdrücke und ihre jeweiligen Entsprechungen in XML.

JSON	XML
`null`	`<Value type="Null"/>`
`true`	`<Value type="Boolean">true</Value>`
`false`	`<Value type="Boolean">false</Value>`
`1`	`<Value type="Number">1</Value>`
`1.0`	`<Value type="Number">1.0</Value>`
`1.2e3`	`<Value type="Number">1200</Value>`
`"Hallo Welt!"`	`<Value type="String">Hallo Welt!</Value>`

`"\u0000"`	escapeStringInXml = true
	`<Value type="String">\u0000</Value>`
	escapeStringInXml = false
	`<Value type="String">&#xFFFD;</Value>`

`{}`	`<Object/>`
`[]`	`<Array/>`

JSON	XML

{
  "key" : "value",
  "a number" : 42,
  "true" : false,
  "emptyList" : [],
  "emptyObject" : {}
}
	<Object>
	  <Value name="key" type="String">value</Value>
	  <Value name="a number" type="Number">42</Value>
	  <Value name="true" type="Boolean">false</Value>
	  <Array name="emptyList" />
	  <Object name="emptyObject" />
	</Object>

[1,2,3]	<Array> <Value type="Number">1</Value> <Value type="Number">2</Value> <Value type="Number">3</Value> </Array>

{
  "title" : "A List",
  "list" : [
  {
  "id" : 1,
  "content" : "first"
  } ,
  {
  "id" : 2,
  "content" : "second"
  } ,
  {
  "id" : 3,
  "content" : "third"
  } ,
  ]
}
	<Object>
	  <Value name="title" type="String">A List</Value>
	  <Array name="list">
	  <Object>
	  <Value name="id" type="Number">1</Value>
	  <Value name="content" type="String">first</Value>
	  </Object>
	  <Object>
	  <Value name="id" type="Number">2</Value>
	  <Value name="content" type="String">second</Value>
	  </Object>
	  <Object>
	  <Value name="id" type="Number">3</Value>
	  <Value name="content" type="String">third</Value>
	  </Object>
	  </Array>
	</Object>

10.20.1.2 JSON Validation

`JSON Validation`: Validiert ein Input-JSON-Dokument gegen ein JSON-Schema. Ist das geprüfte JSON-Dokument gültig, gibt der Adapter dieses unverändert aus. Wenn das geprüfte JSON-Dokument ungültig ist, wird ein XML-Dokument mit einer entsprechenden Fehlermeldung ausgegeben.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`Validate`*: JSON-Dokument validieren

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.json.JSON Validator*: Hauptklasse (Standard)
schemaURL	Pfad zur Schema-Definition (JSON-Schema) **Mögliche Werte:** Beliebige `xstore://`-URL, z. B. *`xstore://Project/Folder/Schema.json`*

Statuswerte

-1 (error)	Während der Ausführung des Adapters trat ein Fehler auf. Nähere Details finden Sie im Server-Log.
1 (successful)	Das geprüfte JSON-Dokument ist gültig gemäß dem angegebenen Schema; Der Adapter gibt das geprüfte Input-JSON-Dokument unverändert aus.
0	Das geprüfte JSON-Dokument ist nicht gültig gemäß dem angegebenen Schema. Der Adapter gibt eine entsprechende Fehlermeldung aus (siehe *Ausgabe*).

Input

Als Input erwartet der Adapter eine JSON-Datei, die er anschließend gegen ein JSON-Schema validiert.

Ausgabe

Der Adapter gibt bei erfolgreicher Validierung das Input-Dokument unverändert aus. Ist das geprüfte JSON-Dokument ungültig, wird ein XML-Dokument mit einer entsprechenden Fehlermeldung ausgegeben, wobei für jeden Fehler ein XML-Element `<Error/>` erzeugt wird. Dieses Element enthält eine detaillierte Fehlerbeschreibung, z.B.:

Beispiel einer Fehlermeldung

```
<ValidationErrors>
    <Error>#/rectangle/a: -5.0 is not higher or equal to 0</Error>
</ValidationErrors>
```

10.20.2 OData Connector

`OData Connector`: Ermöglicht die Kommunikation zu Systemen, die ReST-Services nach OData-Standard bereitstellen z.B. SAP S4/Hana, Microsoft Sharepoint, Microsoft Dynamics Navision und weitere.

Eigenschaften

`Operation`	Vom Adapter ausgeführte Operationen **Mögliche Werte:** • `ReadMetadata`: Ruft das Datenmodell und die Operationen ab, die von einem bestimmten Dienst ausgeführt werden. • `ReadCollections`: Ruft alle verfügbaren Auflistungen aus dem Dienst ab. • `ReadEntitySet`: Ruft eine bestimmte Auflistung aus dem Service ab. • `ReadSingleEntity`: Ruft eine einzelne Entität aus dem Dienst auf, spezifiziert durch die Auflistung und die ID. • `ReadNavigationEntity`: Ruft die zugehörige Auflistung für eine Entität ab. • `ReadAssociation`: Ruft die zugehörige Zuordnung zur Entität ab. • `ReadSingleton`: Ruft das spezifische Singleton ab. • `ReadMediaEntity`: Ruft die Medienentität ab. • `CreateEntity`: Erstellt neue Entitäten im System, spezifiziert durch die Auflistung und die ID in einem XML-Inputdokument. • `UpdateEntity`: Aktualisiert im System vorhandene Entitäten, spezifiziert durch die Auflistung und die ID in einem XML-Inputdokument. • `DeleteEntity`: Löscht im System vorhandene Entitäten, spezifiziert durch die Auflistung und die ID in einem XML-Inputdokument. • `CreateSingleton`: Erstellt ein neues Singleton im System, spezifiziert durch die Auflistung und die ID in einem XML-Inputdokument. • `UpdateSingleton`: Aktualisiert ein Singleton im System, spezifiziert durch die Auflistung und die ID in einem XML-Inputdokument. • `DeleteSingleton`: Löscht ein Singleton im System, spezifiziert durch die Auflistung und die ID in einem XML-Inputdokument. • `CreateMediaEntity`: Erstellt neue Medienentitäten im System, spezifiziert durch die Auflistung und die ID innerhalb der Parameter `entitySet` und `entityId`. • `UpdateMediaEntity`: Aktualisiert Medienentitäten im System, spezifiziert durch die Auflistung und die ID innerhalb der Parameter `entitySet` und `entityId`. • `DeleteMediaEntity`: Löscht Medienentitäten im System, spezifiziert durch die Auflistung und die ID innerhalb der Parameter `entitySet` und `entityId`. • `Functions`: Fragt eine durch den Dienst ausgeführte Operation (Funktion, Aktion) an, wobei die entsprechenden Ausgabewerte abgerufen werden.

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.adapter.odataquery.ODataQueryAdapter*: Hauptklasse (Standard)
`serviceUrl`	OData-Service-URL, z.B. *`https://services.odata.org/V4/TripPinServiceRW`*
`entitySet`	Entitäts-Auflistung

`entityId`	Entitäts-ID
`navigationProperty`	Zur Entität zugehörige Auflistung abrufen
`association`	Zur Entität zugehörige Zuordnung abrufen
`singletonId`	Singleton-ID
`contentTypeMetadata`	ContenType der Service-Metadaten **Mögliche Werte:** • `Full`: Vollständige Metadaten ausgeben • `Minimnal`: Nur grundlegende Metadaten ausgeben • `None`: Keine Metadaten ausgeben

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt und ein Output wurde ausgegeben
`-1`	Es kam zu einem Fehler während der Ausführung und es wurde kein Output ausgegeben

Input

Abhängig von der gewählten Operation, erwartet der Adapter folgende Input-Struktur:

- Die Operationen `ReadaMetadata`, `ReadSingleEntity`, `ReadNavigationEntity`, `ReadSet` und `ReadSingleton` benötigen kein Input-Dokument. Die Parameter `entitySet` und `entityId` werden jedoch benötigt.
- Die Operation `ReadAssociation` benötigen kein Input-Dokument. Die Parameter `entitySet` , `entityId` und `association` werden jedoch benötigt.
- Operation `ReadEntitySet`:
  Folgende Systemabfrageoptionen können für die Operation `ReadEntitySet` verwendet werden:
  - `$count`: Ignoriert `$top`, `$skip`, oder `$expand` Abfrageoptionen und gibt die Gesamtzahl der Ergebnisse auf allen Seiten zurück, einschließlich der Ergebnisse, die mit einem bestimmten `$filter` und einer bestimmten `$search` übereinstimmen.
  - `$skip`: Werden `$top` und `$skip` gemeinsam verwendet, muss `$skip` vor `$top` verwendet unabhängig davon, in welcher Reihenfolge sie in der Abfrage vorkommen.
  - `$top`: Ermöglicht es dem Benutzer eines OData-Dienstes, die maximale Anzahl von Einträgen anzugeben, die von Anfang an zurückgegeben werden sollen.
  - `$orderby`: Beim Anfordern einer Liste von Entitäten aus einem Dienst, liegt es an der Service-Implementierung zu entscheiden, in welcher Reihenfolge sie präsentiert werden.
  - `$filter`: Beim Anfordern einer Liste von Entitäten aus einem Dienst, werden standardmäßig alle Entitäten aus der Liste zurückgegeben. Der Verbraucher eines OData-Dienstes möchte möglicherweise eine Teilmenge empfangen können, indem er bestimmte Kriterien angibt, die jede der zurückgegebenen Entitäten erfüllen muss.

- `$select`: Beim Anfordern einer Entitätsauflistung aus dem Backend, gibt der OData-Dienst eine Liste von Entitäten zurück und jede Entität enthält eine Liste von Eigenschaften.
  - Nur einen Eigenschaftsnamen angeben: `$select=Name`
  - Eine kommagetrennte Liste von Eigenschaften angeben: `$select=Name,Description`
  - Mit einem Stern (`*`) alle Eigenschaften einschließen: `$select=*`
- `$expand`: Zeigt die zugehörigen Entitäten und Stream-Werte an, die inline angegeben werden **müssen**.
- `$search`: Beschränkt das Ergebnis auf die Elemente, die dem angegebenen Suchbegriff entsprechen.
- `$format`: Gibt den Medientyp der Antwort an.

ReadEntitySet

```
<Request>
    <QueryOptions entitySet="Products">
        <Parameter name="format" value="application/xml"></Parameter>
        <Parameter name="search" value="Ergo"></Parameter>
        <Parameter name="top" value="10"></Parameter>
        <Parameter name="skip" value="5"></Parameter>
        <Parameter name="count"></Parameter>
        <Parameter name="filter" value="ID le 10"></Parameter>
        <Parameter name="count" value="true"></Parameter>
        <Parameter name="expand">
            <Items>Category</Items>
        </Parameter>
        <Parameter name="select">
            <Items>ID</Items>
            <Items>Description</Items>
        </Parameter>
    </QueryOptions>
</Request>
```

- Operationen `CreateEntity`, `UpdateEntity` und `DeleteEntity`:

Create-, Update- und Delete-Anfrage

```
<Request>
       <Entity qualifiedName="Microsoft.OData.SampleService.Models.TripPin.Pers
on" entitySet="People">
             <Property name="UserName" value="carlosreyestest" type="String"
isId="true"/>
             <Property name="FirstName" value="Carlos" type="String"/>
             <Property name="LastName" value="Carlos" type="String"/>
             <Collection name="AddressInfo">
                  <Property type="Microsoft.OData.SampleService.Models.TripPin.
Location">
                     <Property name="Address" value="Calle Falsa 123" type="Str
ing"/>
                     <Property name="City" type="Microsoft.OData.SampleService.
Models.TripPin.City">
                           <Property name="CountryRegion" value="Espana" type="
String"/>
                           <Property name="Name" value="Madrid" type="String"/>
                           <Property name="Region" value="Madrid" type="String"
/>
                     </Property>
                  </Property>
             </Collection>
       </Entity>
</Request>
```

- Operationen `CreateSingleton`, `UpdateSingleton` und `DeleteSingleton`:

CreateSingleton-, UpdateSingleton- und DeleteSingleton-Anfrage

```
<Request>
       <Singleton qualifiedName="Microsoft.OData.SampleService.Models.TripPin.P
erson" entitySet="People">
              <Property name="UserName" value="carlosreyestest" type="String"
isId="true"/>
              <Property name="FirstName" value="Carlos" type="String"/>
              <Property name="LastName" value="Carlos" type="String"/>
              <Collection name="AddressInfo">
                   <Property type="Microsoft.OData.SampleService.Models.TripPin.
Location">
                      <Property name="Address" value="Calle Falsa 123" type="Str
ing"/>
                      <Property name="City" type="Microsoft.OData.SampleService.
Models.TripPin.City">
                            <Property name="CountryRegion" value="Espana" type="
String"/>
                            <Property name="Name" value="Madrid" type="String"/>
                            <Property name="Region" value="Madrid" type="String"
/>
                      </Property>
                   </Property>
              </Collection>
       </Singleton>
</Request>
```

- Für die Operationen `CreateMediaEntity`, `UpdateMediaEntity` und `DeleteMediaEntity` sind sowohl die Parameter `entitySet` und `entityId` als auch der Medien-Input im Adapter notwendig.
- Operation `ExecuteFunction`:

Anfrage ExecuteFunction

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
       <Function name="CountCategories">
             <Parameter name="Amount" value="2"></Parameter>
       </Function>
</Request>
```

Unterstütze Typen

Adapter Input-Typ	Typ	Bedeutung
`Binary`	`Edm.Binary`	Binärdaten
`Boolean`	`Edm.Boolean`	Binäre Logik
`Byte`	`Edm.Byte`	Vorzeichenlose 8-Bit Ganzzahl
`Date`	`Edm.Date`	Datum ohne Zeitzonenversatz

Adapter Input-Typ	Typ	Bedeutung
`DateTimeOffset`	`Edm.DateTimeOffset`	Datum und Uhrzeit mit Zeitzonenversatz, keine Schaltsekunden
`Decimal`	`Edm.Decimal`	Zahlenwerte mit dezimaler Darstellung
`Double`	`Edm.Double`	IEEE 754 binary64-Gleitkommazahl (15-17 Dezimalstellen)
`Duration`	`Edm.Duration`	Signierte Dauer in Tagen, Stunden, Minuten und (Teil-)Sekunden
`Guid`	`Edm.Guid`	16 Byte (128 Bit) eindeutige Kennung
`Int16`	`Edm.Int16`	Vorzeichenbehaftete 16-Bit Ganzzahl
`Int32`	`Edm.Int32`	Vorzeichenbehaftete 32-Bit Ganzzahl
`Int64`	`Edm.Int64`	Vorzeichenbehaftete 64-Bit Ganzzahl
`SByte`	`Edm.SByte`	Vorzeichenbehaftete 8-Bit Ganzzahl
`Single`	`Edm.Single`	IEEE 754 binary32-Gleitkommazahl (6-9 Dezimalstellen)
`String`	`Edm.String`	Folge von UTF-8 Zeichen

Ausgabe

Abhängig von der gewählten Operation, gibt der Adapter folgende Strukturen aus:

- Operation `ReadMetadata`:

Antwort ReadMetadata

```
<?xml version="1.0" encoding="UTF-8"?>
<edmx:Edmx Version="4.0"
    xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">
    <edmx:DataServices>
        <Schema xmlns="http://docs.oasis-open.org/odata/ns/edm"
            Namespace="OData.Demo">
            <EntityType Name="Product">
                <Key>
                    <PropertyRef Name="ID" />
                </Key>
                <Property Name="ID" Type="Edm.Int32"></Property>
                <Property Name="Name" Type="Edm.String"></Property>
                <Property Name="Description" Type="Edm.String"></Property>
                <NavigationProperty Name="Category"
                    Type="OData.Demo.Category" Partner="Products"></
NavigationProperty>
            </EntityType>
            <EntityType Name="Category">
                <Key>
                    <PropertyRef Name="ID" />
                </Key>
                <Property Name="ID" Type="Edm.Int32"></Property>
                <Property Name="Name" Type="Edm.String"></Property>
                <NavigationProperty Name="Products"
                    Type="Collection(OData.Demo.Product)" Partner="Category"></
NavigationProperty>
            </EntityType>
            <EntityType Name="Advertisement" HasStream="true">
                <Key>
                    <PropertyRef Name="ID" />
                </Key>
                <Property Name="ID" Type="Edm.Guid"></Property>
                <Property Name="Name" Type="Edm.String"></Property>
                <Property Name="AirDate" Type="Edm.DateTimeOffset"></Property>
            </EntityType>
            <Action Name="Reset" IsBound="false">
                <Parameter Name="Amount" Type="Edm.Int32"></Parameter>
            </Action>
            <Function Name="CountCategories">
                <Parameter Name="Amount" Type="Edm.Int32" Nullable="false"></
Parameter>
                <ReturnType Type="Collection(OData.Demo.Category)" />
            </Function>
            <EntityContainer Name="Container">
                <EntitySet Name="Products" EntityType="OData.Demo.Product">
                    <NavigationPropertyBinding
                        Path="Category" Target="Categories" />
                </EntitySet>
                <EntitySet Name="Categories"
                    EntityType="OData.Demo.Category">
```

```
                    <NavigationPropertyBinding
                        Path="Products" Target="Products" />
                </EntitySet>
                <EntitySet Name="Advertisements"
                    EntityType="OData.Demo.Advertisement"></EntitySet>
                <ActionImport Name="Reset" Action="OData.Demo.Reset"></
ActionImport>
                <FunctionImport Name="CountCategories"
                    Function="OData.Demo.CountCategories"
                    EntitySet="OData.Demo.Categories" IncludeInServiceDocument="
true"></FunctionImport>
            </EntityContainer>
        </Schema>
    </edmx:DataServices>
</edmx:Edmx>
```

- Operationen `ReadSingleEntity` und `ReadSingleton`:

Antwort ReadSingleEntity

```
<?xml version="1.0" encoding="UTF-8"?>
<Object>
    <Value type="String" name="@odata.context">$metadata#Products/$entity</
Value>
    <Value type="Number" name="ID">1</Value>
    <Value type="String" name="Name">Notebook Professional 17</Value>
    <Value type="String" name="Description">Notebook Professional, 2.8GHz - 15
XGA
        - 8GB DDR3 RAM - 500GB</Value>
</Object>
```

- Operation `ReadEntitySet`:

Antwort ReadCollectionEntity

```
<?xml version="1.0" encoding="UTF-8"?>
<Object>
   <Value type="String" name="@odata.context">$metadata#Products</Value>
   <Array name="value">
      <Object>
         <Value type="Number" name="ID">0</Value>
         <Value type="String" name="Name">Notebook Basic 15</Value>
         <Value type="String" name="Description">Notebook Basic, 1.7GHz - 15
XGA - 1024MB DDR2 SDRAM - 40GB</Value>
      </Object>
      <Object>
         <Value type="Number" name="ID">1</Value>
         <Value type="String" name="Name">Notebook Professional 17</Value>
         <Value type="String" name="Description">Notebook Professional, 2.8GHz
- 15 XGA - 8GB DDR3 RAM - 500GB</Value>
      </Object>
      <Object>
         <Value type="Number" name="ID">2</Value>
         <Value type="String" name="Name">1UMTS PDA</Value>
         <Value type="String" name="Description">Ultrafast 3G UMTS/HSDPA Pocket
PC, supports GSM network</Value>
      </Object>
      <Object>
         <Value type="Number" name="ID">3</Value>
         <Value type="String" name="Name">Comfort Easy</Value>
         <Value type="String" name="Description">32 GB Digital Assitant with
high-resolution color screen</Value>
      </Object>
      <Object>
         <Value type="Number" name="ID">4</Value>
         <Value type="String" name="Name">Ergo Screen</Value>
         <Value type="String" name="Description">19 Optimum Resolution 1024 x
768 @ 85Hz, resolution 1280 x 960</Value>
      </Object>
      <Object>
         <Value type="Number" name="ID">5</Value>
         <Value type="String" name="Name">Flat Basic</Value>
         <Value type="String" name="Description">Optimum Hi-Resolution max.
1600 x 1200 @ 85Hz, Dot Pitch: 0.24mm</Value>
      </Object>
   </Array>
</Object>
```

- Operation `ReadCollections`:

Antwort ReadCollections

```
<?xml version="1.0" encoding="UTF-8"?>
<Object>
    <Value type="String" name="@odata.context">$metadata</Value>
    <Array name="value">
        <Object>
            <Value type="String" name="name">Products</Value>
            <Value type="String" name="url">Products</Value>
        </Object>
        <Object>
            <Value type="String" name="name">Categories</Value>
            <Value type="String" name="url">Categories</Value>
        </Object>
        <Object>
            <Value type="String" name="name">Advertisements</Value>
            <Value type="String" name="url">Advertisements</Value>
        </Object>
        <Object>
            <Value type="String" name="name">CountCategories</Value>
            <Value type="String" name="url">CountCategories</Value>
            <Value type="String" name="kind">FunctionImport</Value>
        </Object>
    </Array>
</Object>
```

- Operationen `CreateEntity`, `UpdateEntity` und `DeleteEntity`:

CreateEntity-, UpdateEntity- und DeleteEntity-Antwort

```
<Response>
    <OKEntities entitySet="Products" qualifiedName="OData.Demo.Product">
        <Property isId="true" type="int32" value="1" name="ID"/>
        <Property type="String" value="Car" name="Name"/>
        <Property type="String" value="Good car" name="Description"/>
    </OKEntities>
    <FailEntities entitySet="Products" qualifiedName="OData.Demo.Product">
        <Property isId="true" type="int32" value="2" name="ID"/>
        <Property type="String" value="Cookie" name="Name"/>
        <Property type="String" value="Great cookie" name="Description"/>
    </FailEntities>
</Response>
```

- Operationen `CreateSingleton`, `UpdateSingleton` und `DeleteSingleton`:

CreateSingleton-, UpdateSingleton- und DeleteSingleton-Antwort

```
<Response>
    <OKEntities entitySet="Products" qualifiedName="OData.Demo.Product">
        <Property isId="true" type="int32" value="1" name="ID"/>
        <Property type="String" value="Car" name="Name"/>
        <Property type="String" value="Good car" name="Description"/>
    </OKEntities>
    <FailEntities entitySet="Products" qualifiedName="OData.Demo.Product">
        <Property isId="true" type="int32" value="2" name="ID"/>
        <Property type="String" value="Cookie" name="Name"/>
        <Property type="String" value="Great cookie" name="Description"/>
    </FailEntities>
</Response>
```

- Operation `ExecuteFunction`:

Antwort ExecuteFunction

```
<?xml version="1.0" encoding="UTF-8"?>
<Object>
       <Value type="String" name="@odata.context">$metadata#Collection(OData.De
mo.Category)
       </Value>
       <Array name="value">
             <Object>
                    <Value type="Number" name="ID">0</Value>
                    <Value type="String" name="Name">Notebooks</Value>
             </Object>
             <Object>
                    <Value type="Number" name="ID">1</Value>
                    <Value type="String" name="Name">Organizers</Value>
             </Object>
             <Object>
                    <Value type="Number" name="ID">2</Value>
                    <Value type="String" name="Name">Monitors</Value>
             </Object>
       </Array>
</Object>
```

10.20.3 URL Encoding Converter

`URL Encoding Converter`: Wandelt eine URL oder eine Zeichenkette in eine URL und umgekehrt.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Encode`: Verschlüsselt die angegebene URL • `Decode`: Entschlüsselt die angegebene URL • `EncodePlain`: Verschlüsselt den angegebenen Text • `DecodePlain`: Entschlüsselt den angegebenen Text

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.url.UrlEncoding Converter*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation konnte nicht ausgeführt werden.

Input

Der Adapter erwartet eine XML-Datei mit der zu konvertierenden Zeichenkette bzw. URL als Input.

- `Encode`-Operation:

  Erwartete Struktur für die Operation Encode

  ```
  <?xml version="1.0" encoding="UTF-8" ?>
  <URL>http://www.softproject.de/search?q=Hellö Wörld@X4$</URL>
  ```

- `Decode`-Operation:

  Erwartete Struktur für die Operation Decode

  ```
  <?xml version="1.0" encoding="UTF-8" ?>
  <URL>http://www.softproject.de/search?q=Hell%C3%B6%20W%C3%B6rld@X4$</URL
  ```

- `EncodePlain`-Operation:

  Erwartete Struktur für die Operation EncodePlain

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <Text>Hellö Wörld@X4$</Text>
  ```

- `DecodePlain`-Operation:

Erwartete Struktur für die Operation DecodePlain

```
<?xml version="1.0" encoding="UTF-8"?>
<Text>Hell%C3%B6%20W%C3%B6rld@X4$</Text>
```

Output:

Der Adapter liefert ein XML-Dokument mit der konvertierten URL.

- `Encode`-Operation

Beispiel-Output für die Operation Encode

```
<?xml version="1.0" encoding="UTF-8" ?>
<URL>http://www.softproject.de/search?q=Hell%C3%B6%20W%C3%B6rld@X4$</URL
```

- `Decode`-Operation

Beispiel-Output für die Operation Decode

```
<?xml version="1.0" encoding="UTF-8" ?>
<URL>http://www.softproject.de/search?q=Hellö Wörld@X4$</URL>
```

- `EncodePlain`-Operation:

Beispiel-Output für die Operation EncodePlain

```
<?xml version="1.0" encoding="UTF-8"?>
<Text>Hell%C3%B6%20W%C3%B6rld@X4$</Text>
```

- `DecodePlain`-Operation:

Beispiel-Output für die Operation DecodePlain

```
<?xml version="1.0" encoding="UTF-8"?>
<Text>Hellö Wörld@X4$</Text>
```

10.20.4 URL Reader

`URL Reader`: Liest ein Dokument von einer URL (*Uniform Resource Locator*) und gibt dieses Dokument aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *Read*: Dokument von einer URL lesen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.urlreader.URLReader* : Hauptklasse (Standard)
Url	Uniform Resource Locator (URL) mit beliebigem gültigen Protokoll (`http://`, `ftp://`, `xstore://`, `file://` etc.) **Mögliche Werte:** Beliebige gültige URL (z.B. *https://www.google.com/favicon.ico*)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.20.5 Webservice

10.20.5.1 HTTP(S) Connector

`HTTP(S) Connector`: Kommuniziert mit Servern über das *HyperText Transfer Protocol* (HTTP) oder *HTTP Secure* (HTTPS) auf Basis des *Apache HttpClient* in Version 4. Der Adapter bietet die Standard-HTTP-Methoden `GET`, `POST`, `PUT`, `DELETE`, `HEAD`, `OPTIONS`, `TRACE` und `PATCH`. Zusätzlich können benutzerdefinierte HTTP-Methoden verwendet werden. Dabei unterstützt der Adapter HTTP-Basic-, Digest- und NTLMv2-Authentifizierung sowie Proxy-Server und präemptive Authentifizierung. Zudem unterstützt der Adapter auch Server Name Indication für HTTPS-Verbindungen sowie Client-Zertifikate zur Validierung von Requests.

Ehemalige Bezeichnung bis Version 4.5.1: *HTTP(S) Client 4 Adapter*

Hinweis:

- Im Standardmodus (Parameter `fromXML` deaktiviert) verarbeitet der Adapter je nach Operation ggf. ein beliebiges Input-Dokument mit unverändertem Content Type und Zeichenkodierung.
- Im Expertenmodus (Parameter `fromXML` aktiviert) verarbeitet der Adapter ein spezifisches XML-Dokument, das Header, Cookie-Informationen etc. enthalten kann und mit dem sich die HTTP(S)-Operation detailliert steuern lässt, siehe Parameter `fromXML` und die Beschreibung der Input-Struktur.
- Dieser Adapter löst den bestehenden Adapter *HTTP(S) Communication* ab, siehe HTTP(S) Communication (veraltet).

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`Get`*: Daten von einer URL mit der `GET-Methode` anfordern • *`Post`*: Daten im Body des HTTP-Requests per `POST` an eine URL senden • *`Put`*: Daten im Body des HTTP-Requests über die `PUT`-Methode auf einer URL publizieren • *`Delete`*: die in der HTTP-Anfrage aufgerufene Ressource per `DELETE`-Methode löschen • *`Head`*: Nur die Header-Informationen per `HEAD`-Methode anfordern • *`Options`*: Vom Server per `OPTIONS`-Methode eine Liste der unterstützten HTTP-Methoden abrufen • *`Trace`*: Die vom Server empfangene Anfrage zurückgeben, um ggf. Änderungen von dazwischen liegenden Servern sichtbar zu machen • *`Patch`*: Die im Body des HTTP-Requests abgelegten Daten partiell aktualisieren • *`Execute`*: Daten im Body des HTTP-Request mit einer beliebigen HTTP-Methode senden, siehe auch Input-Beispiel

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.http4.HttpClient4Adapter*: Hauptklasse (Standard)

protocol	Verwendetes Protokoll (HTTP oder HTTPS) **Mögliche Werte:** • `http`: HTTP (unverschlüsselt) verwenden (Standard) • `https`: HTTPS (SSL-verschlüsselt) verwenden
host	IP-Adresse oder Host-Name des physikalischen Servers **Hinweis:** Der Parameter `host` fügt einen `HTTP-Header` namens `host` hinzu. **Mögliche Werte:** Beliebige gültige Zeichenkette ohne Protokollangabe (z. B. *`google.com`*)
vHost	Host-Name eines logischen Servers (*virtual host*), falls dieser durch die physikalische Angabe in Parameter host nicht eindeutig identifiziert werden kann **Hinweis:** Der Parameter `vHost` überschreibt den Wert für den `HTTP-Header` namens `host`, siehe Parameter `host`. Hier können die Ports angegeben werden, ausgenommen der Ports 80 (Standard für HTTP-Verbindungen) und 443 (Standard für HTTPS-Verbindungen). **Mögliche Werte:** Beliebige gültige Zeichenkette ohne Protokollangabe (z. B. *`google.com`*)
port	Portnummer des HTTP(S)-Dienstes **Mögliche Werte:** • Beliebige ganze Zahl (z. B. *`8080`*) • *`-1`*: Für HTTP-Verbindungen Port 80 verwenden; bei HTTPS-Verbindungen Port 443 verwenden (Standard)
path	Ordnerpfad auf dem Server, beginnend mit einem Schrägstrich **Mögliche Werte:** Beliebige Pfadangabe (z. B. */intl/en_ALL/images/logo.gif*)

method	Name der HTTP-Methode, die bei der Operation `Execute` verwendet werden soll. Optional kann die Methode auch über den XML-Request gesetzt werden. **Mögliche Werte:** beliebige Zeichenkette
authenticationEncoding	Zeichenkodierung für die Anmeldedaten **Mögliche Werte:** Angabe einer Zeichenkodierung, z. B. `UTF-8` oder `ISO-8859-1` Mithilfe dieses Parameters können auch Umlaute in den Anmeldedaten verwendet werden.
query	Argument(e) für die HTTP(S)-Anfrage **Mögliche Werte:** Beliebige URL-kodierte Argumente ohne führendes Fragezeichen (z. B. *`Arg1=V1&Arg2=V2`*)
username	Benutzername für eine Basic-, Digest- oder NTLM-Authentifizierung **Mögliche Werte:** • Bei HTTP-Basic-Authentifizierung: der Benutzername • Bei HTTP-Basic-Authentifizierung an einer Windows-Domäne: Zeichenkette nach dem Muster `<Domäne>\<Benutzername>` (z. B. *`MyDomain\MyUserName`*) • Bei NTLM-Authentifizierung an einer Windows-Domäne: nur der Benutzername (und in Parameter ntlmDomain die Domäne angeben) (z. B. *`MyUsername`*)
password	Passwort für eine Basic-, Digest- oder NTLM-Authentifizierung **Mögliche Werte:** Beliebige Zeichenkette
ntlmWorkstation	Workstation (Computer-Name) des Clients (nur für NTLM-Authentifizierung, siehe Parameter `allowNtlm`) **Mögliche Werte:** Beliebiger gültiger Workstation-Name
ntlmDomain	Name der Domäne (nur für NTLM-Authentifizierung, siehe Parameter `allowNtlm`) **Mögliche Werte:** Beliebiger gültiger Domäne-Name (NetBIOS Name)

usePreemptiveBasicAuth	Präemptive Authentifizierung (mit den Zugangsdaten aus Parameter username und password) für HTTP(S)-Aufrufe aktivieren (nur für Basic-Authentifizierung, siehe Parameter allowBasic) **Mögliche Werte:** • `ja`: Zugangsdaten direkt (präemptiv) bei jedem HTTP(S)-Aufruf mitschicken • `nein`: Keine präemptive Authentifizierung (Standard)
allowBasic	HTTP(S)-Basic-Authentifizierung verwenden **Mögliche Werte:** • `ja`: Basic-Authentifizierung verwenden • `nein`: Keine Basic-Authentifizierung verwenden (Standard)
allowDigest	Digest-Authentifizierung verwenden **Mögliche Werte:** • `ja`: Per Digest authentifizieren • `nein`: Nicht per Digest authentifizieren (Standard)
allowNtlm	Proprietäre NTLM-Server-Authentifizierung von Microsoft verwenden **Mögliche Werte:** • `ja`: Per NTLM authentifizieren • `nein`: Nicht per NTLM authentifizieren (Standard)

fromXml	Expertenmodus aktivieren. Der Adapter erwartet ein spezifisches XML-Dokument inklusive Headern, Cookie-Informationen etc., mit dem sich die HTTP(S)-Operation detailliert steuern lässt. **Hinweis:** Im Expertenmodus kann der HTTP-Header `host` ausschließlich über den Parameter `host` gesetzt werden (nicht im XML-Dokument). Alternativ kann der Parameter `host` durch den Parameter `vHost` überschrieben werden, siehe die Parameter `host` bzw. `vHost`. **Mögliche Werte:** • `ja`: Expertenmodus aktivieren (Adapter erwartet ein spezifisches Input-XML-Dokument) • `nein`: Adapter-Input direkt verarbeiten
toXml	Alle Header als Elemente und den Inhalt des Bodys als Base64-kodierten Text in einem XML-Dokument ausgeben **Mögliche Werte:** • `ja`: XML-Dokument ausgeben • `nein`: Server-Antwort unverändert ausgeben (Standard)
hideCookieHeader	Im Ergebnis-XML-Dokument auch die Header-Elemente mit Cookie-Informationen ausgeben (wenn Parameter `toXml` aktiviert ist); Nur zu Entwicklungs- und Testzwecken deaktivieren oder bei Cookies, die vom Standard abweichen **Mögliche Werte:** • `ja`: Keine Cookie-Informationen als Header im Ergebnis-XML-Dokument (Standard) • `nein`: Cookie-Informationen als Header im Ergebnis-XML-Dokument ausgeben
useProxy	Proxy-Server für den Verbindungsaufbau verwenden **Mögliche Werte:** • `ja`: Verbindung über den in Parameter proxyHost angegeben Proxy-Server aufbauen • `nein`: Kein Proxy-Server (Standard)

proxyHost	IP-Adresse oder Host-Name eines Proxy-Servers **Mögliche Werte:** Beliebige gültige Zeichenkette (z. B. *`192.168.1.1`*)
proxyPort	Portnummer des Proxy-Servers **Mögliche Werte:** • Beliebige ganze Zahl • *`-1`*: Standard-Port verwenden (Standard)
proxyUsername	Benutzername für eine Basic-, Digest- oder NTLM-Authentifizierung am Proxy-Server **Mögliche Werte:** • Bei HTTP-Basic-Authentifizierung: der Benutzername • Bei HTTP-Basic-Authentifizierung an einer Windows-Domäne: Zeichenkette nach dem Muster `<Domäne>\<Benutzername>` (z. B. *`MyDomain\MyUserName`*) • Bei NTLM-Authentifizierung an einer Windows-Domäne: nur der Benutzername (und in Parameter `ntlmDomain` die Domäne angeben) (z. B. *`MyUsername`*)
proxyPassword	Passwort für den Proxy-Server **Mögliche Werte:** Beliebige Zeichenkette
proxyNtlmWorkstation	Workstation (Computer-Name), von der die Authentifizierungsanfrage ausgeht (für eine NTLM-Authentifizierung via NTLM-Proxy) **Mögliche Werte:** Beliebiger Workstation-Name
proxyNtlmDomain	Windows-Domäne für eine NTLM-Authentifizierung via NTLM-Proxy **Mögliche Werte:** Beliebiger Domänenname

proxyUsePreemptiveBasicAuth	Präemptive Authentifizierung (mit den Zugangsdaten aus Parameter `proxyUsername` und `proxyPassword`) für den Proxy-Server aktivieren (nur für Basic-Authentifizierung, siehe Parameter `proxyAllowBasic`) **Mögliche Werte:** • `Ja`: Zugangsdaten direkt (präemptiv) bei jedem Proxy-Aufruf mitschicken • `Nein`: Keine präemptive Authentifizierung (Standard)
proxyAllowBasic	Basic-Authentifizierung für den Proxy verwenden **Mögliche Werte:** • `ja`: Basic-Authentifizierung für den Proxy verwenden • `nein`: Keine Basic-Authentifizierung für den Proxy verwenden (Standard)
proxyAllowDigest	Digest-Authentifizierung für den Proxy verwenden **Mögliche Werte:** • `ja`: Am Proxy per Digest authentifizieren • `nein`: Nicht per Digest am Proxy authentifizieren (Standard)
proxyAllowNtlm	Proprietäre NTLM-Server-Authentifizierung von Microsoft für den Proxy verwenden **Mögliche Werte:** • `ja`: Am Proxy per NTLM authentifizieren • `nein`: Nicht per NTLM am Proxy authentifizieren (Standard)
connectTimeout	Timeout-Wert für den Verbindungsaufbau in Millisekunden **Mögliche Werte:** • `>0`: Timeout-Wert in Millisekunden • `<=0`: Kein Timeout (Standard)

socketTimeout	Timeout-Wert zwischen zwei eingehenden Datenpaketen in Millisekunden; Dieser sagt nichts über die Gesamt-Anfragezeit aus. **Mögliche Werte:** • `>0`: Timeout-Wert in Millisekunden • `<=0`: Kein Timeout (Standard)
redirectsEnabled	URL-Weiterleitungen bei einer Server-Antwort mit HTTP-Status *`301 Moved Permanently`* zulassen **Mögliche Werte:** • `ja`: URL-Redirects ausführen (Standard) • `nein`: Keine URL-Redirects ausführen
circularRedirectsEnabled	Zirkuläre Weiterleitungen bei einer Server-Antwort mit HTTP-Status *`301 Moved Permanently`* zulassen **Mögliche Werte:** • `ja`: Zirkuläre Weiterleitungen ausführen (Standard) • `nein`: Keine zirkulären Weiterleitungen ausführen
maxRedirects	Maximale Anzahl der möglichen URL-Weiterleitungen bei einer Server-Antwort mit HTTP-Status *`301 Moved Permanently`* **Mögliche Werte:** • Beliebige ganze Zahl • *`-1`*: Keine Beschränkung (Standard)
acceptAllCertificates	Zertifikatsprüfung abschalten **Mögliche Werte:** • `beliebiger String`: Zertifikate werden auf Vertrauen geprüft (Standard) • `Yes, I want to disable security by accepting untrusted certificates.`: Zertifikatsprüfung wird abgeschaltet **Bitte beachten:** Hier muss die genaue Zeichenfolge eingegeben werden!

acceptAllSslHostnames	Hostname-Überprüfung für Zertifikate abschalten **Mögliche Werte:** • `beliebiger String`: Hostname muss zum Zertifikat passen (Standard) • `Yes, I want to disable security by accepting non-matching hostnames.`: Hostname-Überprüfung wird abgeschaltet **Bitte beachten:** Hier muss die genaue Zeichenfolge eingegeben werden!
useClientCertificate	Client-Zertifikate zur Validierung des Requests verwenden **Mögliche Werte:** • `ja`: Client-Zertifikate zur Validierung des Requests verwenden • `nein`: Kein Client-Zertifikat zur Validierung des Requests verwenden (Standard)
keystoreType	Keystore-Typ für die Speicherung von Client-Zertifikaten (gemäß http://docs.oracle.com/javase/6/docs/technotes/guides/security/StandardNames.html#KeyStore) **Mögliche Werte:** • `jceks`: Proprietäre Keystore-Implementierung von SunJCE verwenden • `jks`: Proprietäre Keystore-Implementierung von SUN verwenden • `pkcs12`: Keystore-Transferformat PKCS12 verwenden
keystoreURL	URL zum Keystore für die Validierung des Requests über Client-Zertifikate **Mögliche Werte:** Beliebige gültige URL, z.B. `file:/C:/Folder/ExampleKeystore`
keystorePassword	Passwort für den Keystore zur Validierung des Requests über Client-Zertifikate **Mögliche Werte:** Beliebige gültige Zeichenkette
keyAlias	Schlüssel-Alias zur Zugriff auf einen bestimmten Schlüssel innerhalb des Keystores **Mögliche Werte:** Beliebige gültige Zeichenkette

aliasPassword	Zugehöriges Passwort für einen bestimmten Schlüssel innerhalb des Keystores **Mögliche Werte:** Beliebige gültige Zeichenkette

Statuswerte

-1	Technischer Fehler: Server antwortet nicht, oder das Adapter-Input-XML kann nicht verarbeitet werden (nur wenn der Expertenmodus über Parameter fromXml aktiviert ist)
>=100	Dreistelliger HTTP-Statuscode je nach Ergebnis des Aufrufs, siehe http://tools.ietf.org/html/rfc2616#section-6.1.1

Input (Standardmodus)

Im Standardmodus (Parameter `fromXML` deaktiviert) verarbeitet der Adapter je nach Operation ggf. ein beliebiges Input-Dokument mit unverändertem Content Type und Zeichenkodierung; siehe Document Properties Changer.

Input (Expertenmodus)

Im Expertenmodus (Parameter `fromXML` aktiviert) verarbeitet der Adapter ein spezifisches XML-Dokument, das Header, Cookie-Informationen etc. enthalten kann und mit dem sich die HTTP(S)-Operation detailliert steuern lässt.

Diese XML-Struktur entspricht im Wesentlichen sowohl der XML-Struktur, die der Adapter ausgibt, wenn Parameter `toXml` aktiviert ist, als auch der *X4 ReST*-Schnittstelle (wenn für die jeweilige URI-Operation in der Konfigurationsdatei `<X4>/X4DB/0/restconfig.xml` in Attribut *`isDirect="false"`* definiert wurde). Einziger Unterschied zum Ergebnis-XML-Dokument von *X4 ReST* ist, dass dieses den Base64-kodierten Body in einem Element `<Content>` (statt `<Body>` beim Adapter) bereitstellt.

In Element `<Cookies>` sind ggf. zudem Cookie-Informationen in jeweils einem Element `<Cookie>` enthalten:

```
<Request>
   <Headers>
      <Header>
         <Name><!--HTTP-Header-Name--></Name>
         <Value><!--Header-Wert--></Value>
      </Header>
      <!-- Oder bei Datumswerten: -->
      <Header>
         <Name><!--HTTP-Header-Name--></Name>
         <Date timezone="Zeitzone"><!--Header-Datumswert--></Date>
      </Header>
   </Headers>
   <Cookies>
      <Cookie expires="Verfallsdatum" path="Pfad" domain="Domäne">
         <Version><!--Versionsnummer--></Version>
         <Secure/>       <!-- Wenn Secure-Attribut gesetzt -->
         <Persistent />  <!-- Bei einem persistenten Cookie -->
         <Domain><!--Domäne--></Domain>
         <Path><!--Pfad--></Path>
         <Name><!--Cookie-Name--></Name>
         <Value><!--Cookie-Wert--></Value>
         <ExpiryDate><!--Verfallsdatum im Format YYYY-MM-DD hh:mm:ss--></ExpiryDate>
      </Cookie>
   </Cookies>
   <Body><!--Base64-kodierte Daten der HTTP-Anfrage--></Body>
</Request>
```

Operation Execute

Bei der Operation `Execute` in Verbindung mit dem aktiviertem Parameter *`fromXML`* kann die HTTP-Methode auch im Input-XML wie folgt gesetzt werden:

```
 <Request method="CUSTOM_METHOD">
   <Headers>
      <!-- ... -->
   </Headers>
   <Cookies>
      <!-- ... -->
   </Cookies>
   <Body><!--Base64-kodierte Daten der HTTP-Anfrage--></Body>
</Request>
```

Ausgabe

Standardmäßig (wenn Parameter `toXml` deaktiviert ist) wird das Ergebnisdokument des HTTP(S)-Aufrufs ausgegeben.

Wenn Parameter `toXml` aktiviert ist, wird ein XML-Dokument ausgegeben, das alle HTTP-Header der Antwort jeweils in einem Element `<Header>` enthält. Falls ein Header-Wert mehrere Werte enthält,

die durch Kommas getrennt sind, so werden diese Teilwerte zusätzlich in jeweils einem Element `<Element>` in den Kindelementen `<Name>` und `<Value>` angegeben.

Der Inhalt des Bodys ist als Base64-kodierte Zeichenkette in Element `<Body>` enthalten. Wenn Parameter `hideCookieHeader` deaktiviert wurde, wird zusätzlich jeder Header ausgegeben, der zum Cookie geführt hat.

```
<Response>
   <Request line="HTTP-Anfragezeile">
      <Method><!--HTTP-Methode--></Method>
      <URI><!--Pfad auf dem Server--></URI>
      <Protocol>HTTP/1.1</Protocol>
   </Request>
   <Status line="HTTP-Statuszeile">
      <Code><!--Dreistelliger Statuscode--></Code>
      <Reason><!--Statusname--></Reason>
      <Protocol><!--HTTP/1.1</Protocol>
   </Status>
   <Headers>
      <Header>
         <Name><!--HTTP-Header-Name--></Name>
         <Value><!--Header-Wert--></Value>
      </Header>
      <!-- Oder bei Datumswerten: -->
      <Header>
         <Name><!--HTTP-Header-Name--></Name>
         <Date timezone="Zeitzone"><!--Header-Datumswert--></Date>
      </Header>
   </Headers>
   <Cookies>
      <Cookie expires="Verfallsdatum" path="Pfad" domain="Domäne">
         <Version><!--Versionsnummer--></Version>
         <Secure/>       <!-- Wenn Secure-Attribut gesetzt -->
         <Persistent />  <!-- Bei einem persistenten Cookie -->
         <Domain><!--Domäne--></Domain>
         <Path><!--Pfad</Path>
         <Name><!--Cookie-Name--></Name>
         <Value><!--Cookie-Wert--></Value>
         <ExpiryDate><!--Verfallsdatum im Format YYYY-MM-DD hh:mm:ss--></ExpiryDate>
      </Cookie>
   </Cookies>
   <Body><!--Base64-kodierte Daten im Body der Antwort--></Body>
</Response>
```

10.20.5.2 Webservice (SOAP) Connector

> **Hinweis**
> Der Webservice (SOAP) Connector wird mit Version 8.0 der X4 BPMS abgekündigt und in zukünftigen Versionen der X4 BPMS nicht mehr unterstützt.

`Webservice (SOAP) Connector`: Sendet eine SOAP-Anfrage (*SOAP Request*) via HTTP(S) zu der URL eines WebServices und gibt die zurückgelieferte SOAP-Antwort (*SOAP Response*) aus. Er kommuniziert mit Servern über HTTP (Hyper Text Transfer Protocol) oder HTTPS (HTTP Secure).

Falls eine Authentifizierung erforderlich ist, muss diese SOAP-basiert über den *SOAP Header* erfolgen; Dessen Aufbau (zertifikats-, token-basiert etc.) muss zunächst mit der Gegenstelle geklärt werden. Zudem werden HTTP- und NTLM-Authentifizierung sowie Proxy-Server unterstützt.

Ehemalige Bezeichnung: *HTTP WebService Request Adapter*

Der Adapter unterstützt *SOAP MTOM* (*Message Transmission Optimization Mechanism*), mit dem sich große Dokumente bzw. Binärdaten in MTOM-basierenden WebServices über XOP (XML-binary Optimized Packaging) performant übertragen lassen, siehe WebService Starter (veraltet).
Die MTOM-Verarbeitung für Anfragen wird nur aktiviert, wenn das Input-Dokument des Adapters den MIME-Typ `text/xml` oder `application/xml` besitzt.
Gesteuert wird die MTOM-Funktion über Processing Instructions, die dem SOAP-Request-Dokument zum Aktivieren von MTOM auf Eingangsseite hinzugefügt werden:

- `<?useMTOM?>`: MTOM-Funktion für diesen SOAP-Request aktivieren
- `<?MTOM-xpath 0-n XPath-Ausdrücke ?>`: Beliebig viele XPath-Ausdrücke, um die Elemente zu definieren, die Binärdaten enthalten
- `<?MTOM-size Größe in KB ?>`: Größe der Daten in Kilobyte, ab der die Komprimierung aktiviert wird (Standard: 1 KB)

Außerdem wandelt der Adapter MTOM-basierende SOAP-Responses, die ein Webservice liefert, automatisch in Standard-SOAP-Responses um.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** `Call`: Adapter ausführen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.httpws.HttpWSAdapter*: Hauptklasse (Standard)
wsURL	URL, an die der SOAP-Request gesendet wird, mit dem Standard-Port *80* **Mögliche Werte:** Beliebige Webservice-URL Abweichende Ports müssen in der URL angegeben werden.
soapAction	HTTP-Header des SOAP-Requests. Der Wert dieses Parameters ist eine URL, die die abgefragte Operation identifiziert.
login	Benutzername für eine einfache HTTP-Authentifizierung **Mögliche Werte:** Beliebige Zeichenkette
pwd	Passwort für eine einfache HTTP-Authentifizierung **Mögliche Werte:** Beliebige Zeichenkette
preemptiveAuth	Präemptive Authentifizierung (mit den Zugangsdaten aus Parameter `login` und `pwd`) für alle HTTP(S)-Aufrufe aktivieren **Mögliche Werte:** • *Ja*: Zugangsdaten direkt (präemptiv) bei jedem HTTP(S)-Aufruf mitschicken • *Nein*: Keine präemptive Authentifizierung (Standard)
pxHost	IP-Adresse oder Host-Name eines Proxy-Servers **Mögliche Werte:** Beliebige gültige Zeichenkette (z.B. *192.168.1.1*)
pxPort	Portnummer des Proxy-Servers **Mögliche Werte:** Beliebige ganze Zahl
pxLogin	Benutzername für den Proxy-Server **Mögliche Werte:** Beliebige Zeichenkette

pxPwd	Passwort für den Proxy-Server **Mögliche Werte:** Beliebige Zeichenkette
pxNTLMDomain	Windows-Domäne für eine NTLM-Authentifizierung via NTLM-Proxy **Mögliche Werte:** Beliebiger Domänenname
pxNTLMhost	Host (Computer-Name), von dem die Authentifizierungsanfrage ausgeht (für eine NTLM-Authentifizierung via NTLM-Proxy) **Mögliche Werte:** Beliebiger Hostname
pxUseNTLM	Proprietäre NTLM-Authentifizierung von Microsoft für den Proxy verwenden **Mögliche Werte:** • *ja*: Per NTLM authentifizieren • *nein*: Nicht per NTLM authentifizieren (Standard)
nTLMDomain	Windows-Domäne für eine NTLM-Server-Authentifizierung **Mögliche Werte:** Beliebiger Domänenname
nTLMhost	Host (Computer-Name), von dem die NTLM-Server-Authentifizierungsanfrage ausgeht **Mögliche Werte:** Beliebiger Hostname
useNTLM	Proprietäre NTLM-Server-Authentifizierung von Microsoft verwenden **Mögliche Werte:** • *ja*: Per NTLM authentifizieren • *nein*: Nicht per NTLM authentifizieren (Standard)
timeout	Zeitbeschränkung für den Verbindungsaufbau in Millisekunden **Mögliche Werte:** • Beliebige ganze Zahl • *0*: Keine Zeitbeschränkung

keystorePath	Kompletter Pfad zur Keystore-Datei; wenn ein Keystore angegeben wird, werden nur Zertifikate dieses Keystores akzeptiert **Mögliche Werte:** • Beliebiger Pfad zur Keystore-Datei (nur Zertifikate dieses Keystores verwenden) • *(kein Pfad)*: Alle Zertifikate akzeptieren
keystorePassword	Passwort für die Keystore-Datei **Mögliche Werte:** Beliebige Zeichenkette

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Um einen Webservice mit MTOM aufzurufen, muss der SOAP-Request die o.g. Processing Instructions enthalten, beispielsweise:

```
<?useMTOM?>
<?MTOM-xpath soapenv:Envelope/soapenv:Body/ws:signVerifyRequest/ws:Service
  RequestData/ws:Document?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ws="http://authentidate.de/ws">
  <soapenv:Header />
  <soapenv:Body>
    <ws:signVerifyRequest>
      <ws:ServiceRequestData>
        <ws:Document>JVBiRi0xLjMKMyAwIG9iago8PC9UeXBlIC9QYWdlCi9QYXJlbnQgM
          SAwIFIKL1Jlc291cmNlcyAyIDAgUgovQ29udGVudHMgNCAwIFI+PgplbmRvYmoKNC
          AwIG9iago8PC9GaWx0ZXIgL0ZsYXRlRGVjb2RlIC9MZW5ndGggMjEzNT4+CnN0cmV
        </ws:Document>
        <ws:DocumentName>R_1639_37480.pdf</ws:DocumentName>
        <ws:ServiceRequest>SIGN</ws:ServiceRequest>
      </ws:ServiceRequestData>
    </ws:signVerifyRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Mehrere Processing Instructions werden immer als einzelne Elemente nacheinander aufgeführt.

10.20.5.3 Webservice Security

`Webservice Security`: Verschlüsselt, signiert und entschlüsselt eine SOAP-Nachricht oder fügt dieser einen Benutzer-Token oder einen Zeitstempel hinzu. Der Adapter erwartet als Input eine gültige SOAP-Nachricht und gibt eine gemäß der gewählten Operation veränderte SOAP-Nachricht aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`Encode`*: Nachricht verschlüsseln • *`Decode`*: Nachricht entschlüsseln • *`Sign Message`*: Nachricht signieren • *`Add Username Token`*: Benutzernamens-Token hinzufügen • *`Add Timestamp`*: Zeitstempel hinzufügen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.wssecurity.WSSecurityAdapter`* : Hauptklasse (Standard)
cryptoProviderClass	Klassennamen für den *WSS4J*-Crypto-Provider (Provider-Klasse muss das Interface `org.apache.ws.security.components.crypto.Crypto` implementieren) **Mögliche Werte:** • Beliebiger Klassenname • *`org.apache.ws.security.components.crypto.Merlin`*: Merlin-Crypto-Provider (Standard)
keystoreType	Keystore-Typ des Crypto-Providers **Mögliche Werte:** • Beliebiger Keystore-Typ • *`jks`*: Java Keystore (Standard)

keystorePassword	Passwort des Keystores **Mögliche Werte:** Beliebige Zeichenkette
keystoreAlias	Alias-Name des Keystores **Mögliche Werte:** Beliebige Zeichenkette
aliasPassword	Alias-Passwort des Keystores **Mögliche Werte:** Beliebige Zeichenkette
keystoreFile	Pfad zur Keystore-Datei
encodingAlgorithm	Algorithmus für die symmetrische Verschlüsselung einer Nachricht **Mögliche Werte:** • *`Triple DES`*: *Data Encryption Standard* -Dreifachverschlüsselungs-Algorithmus verwenden • *`AES 128`*: *Advanced Encryption Standard* -Algorithmus mit 128 Bit Schlüssellänge verwenden • *`AES 192`*: *Advanced Encryption Standard* -Algorithmus mit 192 Bit Schlüssellänge verwenden • *`AES 256`*: *Advanced Encryption Standard* -Algorithmus mit 256 Bit Schlüssellänge verwenden
canonicalization	Kanonisierungs-Algorithmus für die Enkodierung/Signatur gemäß http://www.w3.org/TR/xml-exc-c14n/ oder http://www.w3.org/TR/xml-c14n
keyEncodingAlgorithm	Symmetrischer Verschlüsselungs-Algorithmus für den Schlüssel **Mögliche Werte:** • *`RSA15`*: RSA 15 • *`RSA OEP`*: RSA OEP

keyType	Schlüssel-ID-Typ gemäß *WS Security X.509 certificate token profile* **Mögliche Werte:** • `ISSUER SERIAL`: Aussteller und Seriennummer • `BST DIRECT REFERENCE`: Binärer Sicherheits-Token • `SKI KEY IDENTIFIER`: Subject Key Identifier • `X509 KEY IDENTIFIER`: X.509-Aussteller und Seriennummer
signatureAlgorithm	Signatur-Algorithmus **Mögliche Werte:** • `RSA with SHA`: RSA mit SHA verwenden • `DSA with SHA1 (DSS)`: DSA mit SHA1 (DSS) verwenden
actor	Von der Sicherheitsengine verwendeter Aktor/Rolle gemäß SOAP-Spezifikation
keyAlias	Schlüssel- oder Zertifikats-Alias im Keystore **Mögliche Werte:** Beliebige Zeichenkette
keyPassword	Passwort für den Keystore-Alias **Mögliche Werte:** Beliebige Zeichenkette
parts	Nachrichten-Teile, die verschlüsselt oder signiert werden sollen **Mögliche Werte:** Zeichenkette mit Element-Name, Namensraum und der Verschlüsselungsmethode in Anführungszeichen, durch Semikolon getrennt **Beispiel:** `"Element";"";"Content"`
embededName	Embedded Name **Mögliche Werte:** Beliebige Zeichenkette

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.20.6 X4 Streaming Support

? Unknown Attachment `X4 Streaming Support`: Nimmt ein XML oder XML-Fragment (well-nested) und erzeugt daraus ein Dokument, das als Ausgabe für den ReST Starter verwendet werden kann. Innerhalb dieses XML(-Fragments) wird nach `Stream`-Elementen im Namensraum `x4feature:reststream` gesucht. Diese werden in die Ausgabe übernommen, während alle anderen Teile Base64-kodiert werden. Somit lassen sich XML-Dokumente als ReST-Ausgabe erzeugen in denen ein Stream Base64-kodiert eingebettet ist (z. B. eine SOAP-Response, wenn nicht mit MTOM gearbeitet wird).

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** `convertXML`: XML konvertieren

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.reststreamsupport.RestStreamSupportAdapter*: Hauptklasse (Standard)
inputEncoding	Zeichenkodierung des Input-Dokuments **Mögliche Werte:** Beliebige gültige Zeichenkodierung (z. B. *UTF-8* (Standard))

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Der Adapter-Input kann wie folgt aufgebaut sein:

```
<?xml version="1.0" encoding="UTF-8"?>
<RootElement>
    <SomeElementBefore>Text</SomeElementBefore>
    <ContainerForTheStream>
        <Stream xmlns="x4feature:reststream" url="xstore://SomeProject/someData.xml"
transcoding="base64-UTF-8"/>
    </ContainerForTheStream>
    <SomeElementAfter>More Text</SomeElementAfter>
</RootElement>
```

Ausgabe

Folgende XML-Struktur wird vom Adapter ausgegeben:

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
    <Status code="200"/>
    <Headers/>
    <Content encoding="base64">PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPF
Jvb3RFbGVtZW50PgoJPFNv
    bWVFbGVtZW50QmVmb3JlPlReHQ8L1NvbWVFbGVtZW50QmVmb3JlPgoJPENvbnRhaW5lckZvclRo
    ZVN0cmVhbT4=<Stream xmlns="x4feature:reststream" url="xstore://SomeProject/
someData.xml" transcoding="base64-UTF-8"></Stream>
    PC9Db250YWluZXJGb3JUaGVTdHJlYW0+Cgk8U29tZUVsZW1lbnRBZnRlcj5Nb3JlIFRleHQ8L1Nv
    bWVFbGVtZW50QWZ0ZXI+CjwvUm9vdEVsZW1lbnQ+
    </Content>
</Response>
```

Diese Ausgabe kann sofort an den ReST-Starter zurückgegeben werden, oder erst noch mit entsprechenden Headern versehen werden.

Inhalt von xstore://SomeProject/someData.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Hallo><Welt/></Hallo>
```

Ergebnis der ReST-Antwort

```
<?xml version="1.0" encoding="UTF-8"?>
<RootElement>
    <SomeElementBefore>Text</SomeElementBefore>
    <ContainerForTheStream>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxIYW
xsbz48V2VsdC8+PC9IYWxsbz4=</ContainerForTheStream>
    <SomeElementAfter>More Text</SomeElementAfter>
</RootElement>
```

Derzeit gibt es keine Unterstützung wenn ein Stream in ein Attribut eingesetzt werden soll. In diesem Fall muss das entsprechende Element und Attribut mit Base64-Strings im `Content`-Element zusammengebaut werden.

10.20.7 XOP (MTOM)

Unknown Attachment `XOP (MTOM)`: Erzeugt XOP/MTOM-Pakete aus XML-Dokumenten, löst XOP-Referenzen aus einem XOP/MTOM-Paket auf und gibt ein XML-Dokument aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`CleanupXml`*: Entfernt `<?xop-candidate?>`-Anweisungen und normalisiert deren Elternelement • *`MakeSimplepart`*: Entfernt `<?xop-candidate?>`-Anweisungen und normalisiert deren Elternelement • *`MakeMultipart`*: aus einem XML-Dokument ein XOP/MTOM-Paket erzeugen; Die Ausgabestruktur ist ein MIME-XML (siehe auch MIME Multipart) • *`UnpackMultipart`*: XOP-Referenzen aus einem MIME-XML, welches ein XOP/MTOM-Paket darstellt, auflösen; Ausgabe ist ein XML-Dokument (die effektive Nutzlast)

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.xop.Adapter*: Hauptklasse (Standard)
RequiredMinimumPartSavings	Anzahl der Bytes, die pro Element eingespart werden müssen, damit ein Element via XOP/MTOM ausgelagert wird • Beliebige ganze Zahl, z.B. *300* • `0` (Standard) • `<0`: Ineffizienzen werden akzeptiert, d.h. es wird akzeptiert, dass mehr Bytes übertragen werden, als wenn nicht ausgelagert wird • `<= -300`: Die Auslagerung soll unabhängig von den Ersparnissen stattfinden (Auslagern hat einen Overhead von ca. 260–270 Bytes pro Auslagerung)

RequiredMinimumTotalSavings	Legt fest wie viele Bytes insgesamt durch Auslagerungen eingespart werden müssen, damit die Auslagerungen durchgeführt werden. • `0` (Standard) • `<0`: Das Ergebnis-MIME-Dokument darf größer sein als ein Ergebnisdokument, für welches keine Auslagerungen durchgeführt wurden. Mit diesem Parameter kann ein hoher negativer Wert in `RequiredMinimumPartSavings` relativiert werden.

Statuswerte

1	Die Operation wurde ohne Probleme ausgeführt
`0`	Die Operation wurde nicht oder nicht vollständig ausgeführt
-1	Fehler bei der Verarbeitung; Wird typischerweise durch ungültige/unvollständige Input-Dokumente ausgelöst.

10.20.7.1 Operation CleanupXml

Die Adapter-Operation `CleanupXml` führt die Normalisierungsarbeiten für die Erstellung eines XOP-Pakets durch. Dabei wird weder ein Kandidat (identifiziert durch `<?xop-candidate?>`) ausgelagert noch wird ein Mime-XML-Dokument erzeugt.

Der `mimetype` des resultierenden X4-Dokuments (siehe Document Properties Changer) wird wie folgt ermittelt.

Eingehender *mimetype*	Eingehendes *encoding*	XML-Dokument ist ...	resultierender *mimetype*	resultierendes *encoding*
leer	*beliebig*	kein SOAP	`text/xml`	`UTF-8`
`some/type`	*beliebig*	kein SOAP	`some/type`	`UTF-8`
beliebig	*beliebig*	SOAP 1.1	`text/xml`	`UTF-8`
beliebig	*beliebig*	SOAP 1.2	`application/soap+xml`	`UTF-8`

Beispiel

Input

```
<SomeXml>
    <blob1>
        <?xop-candidate?>
            SGFs
            bG8=
    </blob1>
    <blob2>
        <?xop-candidate text/plain?>
            SGFs
            bG8g
            SGFs
            bG8=
    </blob2>
</SomeXml>
```

Output

```
<SomeXml>
    <blob1>SGFsbG8=</blob1>
    <blob2>SGFsbG8gSGFsbG8=</blob2>
</SomeXml>
```

10.20.7.2 Operation MakeSimplepart

Die Adapter-Operation `MakeSimplepart` führt die Normalisierungsarbeiten für die Erstellung eines XOP-Pakets durch, wobei kein Kandidat (identifiziert durch `<?xop-candidate?>`) ausgelagert wird. Das Ergebnis ist ein Mime-XML-Dokument mit nur einem Part (kein Multipart).

Der Mime-Part hat zwei Header:

- `content-transfer-encoding: binary`
- `content-type:` Wert siehe unten

Eingehender *mimetype*	Eingehendes *encoding*	XML-Dokument ist ...	resultierender `content-type` Header
leer	*beliebig*	kein SOAP	`text/xml; charset=UTF-8`
`some/type`	*beliebig*	kein SOAP	`some/type; charset=UTF-8`
beliebig	*beliebig*	SOAP 1.1	`text/xml; charset=UTF-8`
beliebig	*beliebig*	SOAP 1.2	`application/soap+xml; charset=UTF-8`

Beispiel

Input

```
<SomeXml>
    <blob1>
        <?xop-candidate?>
            SGFs
            bG8=
    </blob1>
    <blob2>
        <?xop-candidate text/plain?>
            SGFs
            bG8g
            SGFs
            bG8=
    </blob2>
</SomeXml>
```

Output

```
<Part>
    <Header>
        <Name>content-type</Name>
        <Value>text/xml; charset=UTF-8</Value>
        <Type>text/xml</Type>
        <Parameter>
            <Name>charset</Name>
            <Value>UTF-8</Value>
        </Parameter>
    </Header>
    <Header>
        <Name>content-transfer-encoding</Name>
        <Value>binary</Value>
    </Header>
    <Content encoding="base64">PD94bWwgdmVyc2lvb... more base64 ...g0K</Content>
</Part>
```

10.20.7.3 Operation MakeMultipart

Die Adapter-Operation *MakeMultipart* wird verwendet, um die Übertragung von XML-Dokumenten mit größeren Binär(Base64)-Inhalten zu optimieren. Haupteinsatzgebiet sind SOAP-Webservices, aber auch jede andere XML.

Input: Beispiel einer Input-XML für die Operation MakeMultipart

```
<SomeXml>
    <blob1>
        <?xop-candidate?>
            SGFsbG8=
    </blob1>
    <blob2>
        <?xop-candidate text/plain?>
            SGFsbG8gSGFsbG8=
    </blob2>
</SomeXml>
```

Erläuterung

- Mit der XML-Verarbeitungsanweisung `<?xop-candidate?>` wird angegeben, welcher Textknoten ausgelagert werden soll
- Wird ein Medien-Typ als Parameter angegeben, z.B.
  `<?xop-candidate text/plain?>`
  , so wird dieser Medien-Typ verwendet
- Wird kein Medien-Typ als Parameter angegeben, so wird immer `application/octet-stream` als Medien-Typ angenommen
- Über die Adapter-Parameter `RequiredMinimumPartSavings` und `RequiredMinimumTotalSavings` kann gesteuert werden, wie groß die Ersparnisse sein müssen, damit eine Auslagerung aus dem Input-XML stattfindet.

Output: Beispiel eines erzeugten XML-Dokuments mit MakeMultipart

```
<Multipart>
    <Header>
        <Name>content-transfer-encoding</Name>
        <Value>binary</Value>
    </Header>
    <Header>
        <Name>content-type</Name>
        <Value>multipart/related; type="application/xop+xml";
            start="&lt;0@2eba3b7d-7eab-448d-be94-e5dd2bc6e902&gt;";
            start-info="application/xml";
            boundary=3f032702-07f2-46fa-8108-f2258033727e</Value>
        <Type>multipart/related</Type>
        <Parameter>
            <Name>type</Name>
            <Value>application/xop+xml</Value>
        </Parameter>
        <Parameter>
            <Name>start</Name>
            <Value>&lt;0@2eba3b7d-7eab-448d-be94-e5dd2bc6e902&gt;</Value>
        </Parameter>
        <Parameter>
            <Name>start-info</Name>
            <Value>application/xml</Value>
        </Parameter>
        <Parameter>
            <Name>boundary</Name>
            <Value>3f032702-07f2-46fa-8108-f2258033727e</Value>
        </Parameter>
    </Header>
    <Prolog />
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>application/xop+xml; charset=UTF-8; type="application/xml"
            </Value>
            <Type>application/xop+xml</Type>
            <Parameter>
                <Name>charset</Name>
                <Value>UTF-8</Value>
            </Parameter>
            <Parameter>
                <Name>type</Name>
                <Value>application/xml</Value>
            </Parameter>
        </Header>
        <Header>
            <Name>content-transfer-encoding</Name>
            <Value>binary</Value>
        </Header>
        <Header>
            <Name>content-id</Name>
```

```
            <Value>&lt;0@2eba3b7d-7eab-448d-be94-e5dd2bc6e902&gt;</Value>
            <Id asUrl="cid:0@2eba3b7d-7eab-448d-be94-e5dd2bc6e902">
                0@2eba3b7d-7eab-448d-be94-e5dd2bc6e902
            </Id>
        </Header>
        <Content encoding="base64">PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz
4NCjxTb21lWG1sPgoJPGJsb2IxPjx4b3A6SW5jbHVkZSBocmVmPSJjaWQ6MUAxOGE0ODI3ZC1lZWY3LTQ5NjU
tYmY3OS1lN2YwZDBjYTI5OTQiIHhtbG5zOnhvcD0iaHR0cDovL3d3dy53My5vcmcvMjAwNC8wOC94b3AvaW5j
bHVkZSIvPjwvYmxvYjE+Cgk8YmxvYjI+PHhvcDpJbmNsdWRlIGhyZWY9ImNpZDoyQDkyYzhlNjM5LTkwZjktN
DJjNC1hMGNhLTNkOTdiODQzZTQ4YSIgeG1sbnM6eG9wPSJodHRwOi8vd3d3LnczLm9yZy8yMDA0LzA4L3hvcC
9pbmNsdWRlIi8+PC9ibG9iMj4KPC9Tb21lWG1sPg0K</Content>
    </Part>
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>application/octet-stream</Value>
            <Type>application/octet-stream</Type>
        </Header>
        <Header>
            <Name>content-transfer-encoding</Name>
            <Value>binary</Value>
        </Header>
        <Header>
            <Name>content-id</Name>
            <Value>&lt;1@18a4827d-eef7-4965-bf79-e7f0d0ca2994&gt;</Value>
            <Id asUrl="cid:1@18a4827d-eef7-4965-bf79-e7f0d0ca2994">
                1@18a4827d-eef7-4965-bf79-e7f0d0ca2994
            </Id>
        </Header>
        <Content encoding="base64">SGFsbG8=</Content>
    </Part>
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>text/plain</Value>
            <Type>text/plain</Type>
        </Header>
        <Header>
            <Name>content-transfer-encoding</Name>
            <Value>binary</Value>
        </Header>
        <Header>
            <Name>content-id</Name>
            <Value>&lt;2@92c8e639-90f9-42c4-a0ca-3d97b843e48a&gt;</Value>
            <Id asUrl="cid:2@92c8e639-90f9-42c4-a0ca-3d97b843e48a">
                2@92c8e639-90f9-42c4-a0ca-3d97b843e48a
            </Id>
        </Header>
        <Content encoding="base64">SGFsbG8gSGFsbG8=</Content>
    </Part>
    <Epilog />
</Multipart>
```

Dieses Output-XML entspricht einem MIME-Multipart-Dokument in dem sich als erster Part das modifizierte Eingangs-XML befindet, und alle weiteren Parts die extrahierten Base64-Einschlüsse des Input-XMLs sind.

Output: Entsprechendes MIME-Multipart-Dokument

```
content-transfer-encoding: binary
content-type: multipart/related; type="application/xop+xml";
start="<0@2eba3b7d-7eab-448d-be94-e5dd2bc6e902>"; start-info="application/xml";
boundary=3f032702-07f2-46fa-8108-f2258033727e

--3f032702-07f2-46fa-8108-f2258033727e
content-type: application/xop+xml; charset=UTF-8; type="application/xml"
content-transfer-encoding: binary
content-id: <0@2eba3b7d-7eab-448d-be94-e5dd2bc6e902>

<SomeXml>
    <blob1><xop:Include href="cid:1@18a4827d-eef7-4965-bf79-e7f0d0ca2994"
xmlns:xop="http://www.w3.org/2004/08/xop/include"/></blob1>
    <blob2><xop:Include href="cid:2@92c8e639-90f9-42c4-a0ca-3d97b843e48a"
xmlns:xop="http://www.w3.org/2004/08/xop/include"/></blob2>
</SomeXml>

--3f032702-07f2-46fa-8108-f2258033727e
content-type: application/octet-stream
content-transfer-encoding: binary
content-id: <1@18a4827d-eef7-4965-bf79-e7f0d0ca2994>

Hallo
--3f032702-07f2-46fa-8108-f2258033727e
content-type: text/plain
content-transfer-encoding: binary
content-id: <2@92c8e639-90f9-42c4-a0ca-3d97b843e48a>

Hallo Hallo
--3f032702-07f2-46fa-8108-f2258033727e--
```

Wenn keine Auslagerungen stattgefunden haben, weil z.B. keine Markierungen existieren, oder nicht genug Ersparnisse zusammen gekommen sind, liefert der Adapter den Status `0` für diese Operation.

10.20.7.4 Operation UnpackMultipart

Die Adapter-Operation *`UnpackMultipart`* entspricht der inversen Operation zu *`MakeMultipart`*. Sie löst aus einem MIME-XML, das ein XOP/MTOM-Paket darstellt, die `xop:Include` Elemente auf. Als Ergebnis kommt das Nutzlast-XML des XOP-Pakets (z.B. der SOAP-Envelope) heraus, wobei die `xop:Include` Elemente durch die entsprechenden Base64-Strings ersetzt wurden.

Die Operation akzeptiert auch ein MIME-XML, welches nur ein Part ist. In diesem Fall wird das Part einfach nur ausgepackt. Dann wird jedoch nicht garantiert, dass das Ergebnis ein XML ist, z.B. wird ein Part reinschicken das ein PDF darstellt, ist das Ergebnis das PDF.

Bei den `xop:Include`-Referenzen wird nur das URL-Protokoll `cid:` unterstützt. Die so referenzierten Teile müssen sich im selben Multipart befinden wie das Root-Part, d.h. sie müssen Geschwister-Knoten des Root-Parts sein.

Input: Beispiel Input für die Operation UnpackMultipart

```
<Multipart>
    <Header>
        <Name>content-transfer-encoding</Name>
        <Value>binary</Value>
    </Header>
    <Header>
        <Name>content-type</Name>
        <Value>multipart/related; type="application/xop+xml";
            start="&lt;0@2eba3b7d-7eab-448d-be94-e5dd2bc6e902&gt;";
            start-info="application/xml";
            boundary=3f032702-07f2-46fa-8108-f2258033727e</Value>
        <Type>multipart/related</Type>
        <Parameter>
            <Name>type</Name>
            <Value>application/xop+xml</Value>
        </Parameter>
        <Parameter>
            <Name>start</Name>
            <Value>&lt;0@2eba3b7d-7eab-448d-be94-e5dd2bc6e902&gt;</Value>
        </Parameter>
        <Parameter>
            <Name>start-info</Name>
            <Value>application/xml</Value>
        </Parameter>
        <Parameter>
            <Name>boundary</Name>
            <Value>3f032702-07f2-46fa-8108-f2258033727e</Value>
        </Parameter>
    </Header>
    <Prolog />
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>application/xop+xml; charset=UTF-8; type="application/xml"
            </Value>
            <Type>application/xop+xml</Type>
            <Parameter>
                <Name>charset</Name>
                <Value>UTF-8</Value>
            </Parameter>
            <Parameter>
                <Name>type</Name>
                <Value>application/xml</Value>
            </Parameter>
        </Header>
        <Header>
            <Name>content-transfer-encoding</Name>
            <Value>binary</Value>
        </Header>
        <Header>
            <Name>content-id</Name>
```

```
            <Value>&lt;0@2eba3b7d-7eab-448d-be94-e5dd2bc6e902&gt;</Value>
            <Id asUrl="cid:0@2eba3b7d-7eab-448d-be94-e5dd2bc6e902">
                0@2eba3b7d-7eab-448d-be94-e5dd2bc6e902
            </Id>
        </Header>
        <Content encoding="base64">PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz
4NCjxTb21lWG1sPgoJPGJsb2IxPjx4b3A6SW5jbHVkZSBocmVmPSJjaWQ6MUAxOGE0ODI3ZC1lZWY3LTQ5NjU
tYmY3OS1lN2YwZDBjYTI5OTQiIHhtbG5zOnhvcD0iaHR0cDovL3d3dy53My5vcmcvMjAwNC8wOC94b3AvaW5j
bHVkZSIvPjwvYmxvYjE+Cgk8YmxvYjI+PHhvcDpJbmNsdWRlIGhyZWY9ImNpZDoyQDkyYzhlNjM5LTkwZjktN
DJjNC1hMGNhLTNkOTdiODQzZTQ4YSIgeG1sbnM6eG9wPSJodHRwOi8vd3d3LnczLm9yZy8yMDA0LzA4L3hvcC
9pbmNsdWRlIi8+PC9ibG9iMj4KPC9Tb21lWG1sPg0K
        </Content>
    </Part>
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>application/octet-stream</Value>
            <Type>application/octet-stream</Type>
        </Header>
        <Header>
            <Name>content-transfer-encoding</Name>
            <Value>binary</Value>
        </Header>
        <Header>
            <Name>content-id</Name>
            <Value>&lt;1@18a4827d-eef7-4965-bf79-e7f0d0ca2994&gt;</Value>
            <Id asUrl="cid:1@18a4827d-eef7-4965-bf79-e7f0d0ca2994">
                1@18a4827d-eef7-4965-bf79-e7f0d0ca2994
            </Id>
        </Header>
        <Content encoding="base64">SGFsbG8=</Content>
    </Part>
    <Part>
        <Header>
            <Name>content-type</Name>
            <Value>text/plain</Value>
            <Type>text/plain</Type>
        </Header>
        <Header>
            <Name>content-transfer-encoding</Name>
            <Value>binary</Value>
        </Header>
        <Header>
            <Name>content-id</Name>
            <Value>&lt;2@92c8e639-90f9-42c4-a0ca-3d97b843e48a&gt;</Value>
            <Id asUrl="cid:2@92c8e639-90f9-42c4-a0ca-3d97b843e48a">
                2@92c8e639-90f9-42c4-a0ca-3d97b843e48a
            </Id>
        </Header>
        <Content encoding="base64">SGFsbG8gSGFsbG8=</Content>
    </Part>
    <Epilog />
</Multipart>
```

Output ist ein XML (oder ein beliebiges Dokument, wenn ein MIME-Part-XML hineingeschickt wird)

Output: Beispiel einer Output-XML

```
<SomeXml>
    <blob1>SGFsbG8=</blob1>
    <blob2>SGFsbG8gSGFsbG8=</blob2>
</SomeXml>
```

Wenn nicht alle `xop:Include`-Element ersetzt werden konnten, weil die Referenzen nicht aufgelöst werden konnten, gibt der Adapter den Status `0` zurück und die Ausgabe ist das Nutzlast-XML mit allen Ersetzungen, die durchgeführt werden konnten.

Ein Status von `-1` wird verwendet, um verschiedene Fehler in dem MIME-XML anzuzeigen.

10.20.8 YAML Converter

`YAML Converter`: Konvertiert ein YAML-Dokument in ein XML-Dokument und umgekehrt.

Eigenschaften

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • `Convert YAML to XML`: Konvertiert ein YAML-Dokument in ein XML-Dokument • `Convert XML to YAML`: Konvertiert ein XML-Dokument in ein YAML-Dokument

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.adapter.yaml.YamlAdapter*: Hauptklasse (Standard)
`processStrings`	Definiert, ob numerische Werte und boolean Werte vom Adapter gelesen und normiert werden (*`true`*) oder als literale Zeichenketten behandelt werden (*`false`*). **Mögliche Werte:** • `true`: Numerische Werte und boolean Werte werden gelesen und normiert • `false`: Numerische Werte und boolean Werte werden als literale Zeichenketten behandelt

Statuswerte

`1`	Die Konvertierung war erfolgreich und eine Ausgabe wird erzeugt
`-1`	Während der Konvertierung kam es zu einem Fehler und es wird keine Ausgabe erzeugt

Input

Je nach gewählter Operation erwartet der Adapter folgende Input-Struktur:

- `Convert YAML to XML`:

  YAML Input

  ```yaml
  example:
      element1:
      - value1
      - value2
      - childValue1: 1
        childValue2: 2
      element2: value3
  ```

- `Convert XML to YAML`:

  XML Input

  ```xml
  <Object>
      <Object name="example">
         <Array name="element1">
           <Value>value1</Value>
           <Value>value2</Value>
           <Object>
            <Value name="childValue1">1</Value>
            <Value name="childValue2">2</Value>
           </Object>
         </Array>
         <Value name="element2">value3</Value>
      </Object>
  </Object>
  ```

Ausgabe

Je nach gewählter Operation gibt der Adapter folgend Struktur aus:

- `Convert YAML to XML`:

XML Output

```
<Object>
    <Array name="element1">
       <Value>value1</Value>
       <Value>value2</Value>
       <Object>
         <Value name="childValue1">1</Value>
         <Value name="childValue2">2</Value>
       </Object>
    </Array>
    <Value name="element2">value3</Value>
</Object>
```

- Convert XML to YAML:

YAML Output

```
element1:
- value1
- value2
- {childValue1: 1, childValue2: 2}
element2: value3
```

10.21 X4 API

- Deployments
- Global Parameter Management
- Keycloak Management
- Process Monitoring
- Process Monitoring Management
- Process Runtime Management
- Project Parameter Management
- Savepoint Management
- Schedule Management
- System Configuration Management
- Third Party License Summary
- Web App Configuration Management
- Web App Theme Management

10.21.1 Deployments

`Deployments`: Listet die auf dem Server bereitgestellten Prozesse und Ressourcen im X4 Repository auf. Die Auflistung beinhaltet auch die mithilfe von X4 Apps bereitgestellten Prozesse.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `List`: Listet alle bereitgestellten Prozesse und Ressourcen in einem XML-Dokument auf. • `Get Image`: Gibt ein Bild des Prozess-Workflows aus. • `Get Image Metadata`: Gibt die Metadaten eines Bildes des Prozess-Workflows aus. • `Reload Repository`: Gibt eine Liste aller Projekte und den Repository-Typ für jedes Element aus. • `Reload Project`: Gibt das Projekt und den Typ des zugehörigen Repositorys für jedes Element aus.

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.adapter.deployed.app.DeployedAppAdapter (Standard)*
`repository`	Repository-Name
`project`	Projektname
`processPath`	Prozesspfad und -name

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

ⓘ Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

10.21.1.1 Beispiele

10.21.1.1.1 Output der Operation List

Der Adapter erzeugt ein XML-Dokument als Output.

```
<DeployedAppResponse>
    <Repository name="1">
        <Project name="MyProject" type="ESB" uuid="695245fa-3b07-4e0e-b9d0-
b53fdcb45e51">
            <Resource path="Adapters\MyAdapter.fa"/>
            <Resource path="pom.xml"/>
            <Resource path="Processes\MyProcess.wrf"/>
            <Resource path="Resources"/>
            <Resource path="TemporaryFiles"/>
            <Resource path="Transformations"/>
            <Resource path="MyProject.wsinc"/>
        </Project>
    </Repository>
</DeployedAppResponse>
```

10.21.1.1.2 Output der Operation Get Image Metadata

```
<ProcessElements>
    <ProcessElement id="0" name="Start" x="17" y="119" height="32" width="32"/>
    <ProcessElement id="1" name="Stop" x="476" y="119" height="32" width="32"/>
    <ProcessElement id="2" name="Processes_Deployed.fa" x="204" y="119" height="32"
 width="32" path="ESBProject_1/Processes/Deployment/ProcessImage-14057/
process_processImageWRF.wrf/Processes_Deployed.fa"/>
</ProcessElements>
```

10.21.1.1.3 Output der Operation Reload Project

```
<GetProjectResponse>
    <Project name="AMQP_Project" repository="1" isLibrary="false" projectType="ESB"
 projectVersion="7.2.0-SNAPSHOT" projectId="fd1a244b-245b-4c81-adb1-22499e57aa5a">
        <Properties>
            <Properties>
                <ProjectProperty>
                    <Key>disableWebService</Key>
                    <Value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:boolean">false</Value>
                </ProjectProperty>
            </Properties>
        </Properties>
        <RepositoryPaths>
            <RepositoryPath>
                <repository>1</repository>
                <project>AMQP_Project</project>
                <projectSegments>Adapters</projectSegments>
                <projectSegments>AMQP_Publisher_DE.fa</projectSegments>
                <lastProjectSegmentIndex>1</lastProjectSegmentIndex>
                <isDirectory>false</isDirectory>
                <repositoryMembership>INCLUDED</repositoryMembership>
            </RepositoryPath>
        </RepositoryPaths>
    </Project>
</GetProjectResponse>
```

10.21.1.1.4 Output der Operation Reload Repository

```
<GetProjectsResponse>
    <Project name="ESB_XVIER-14717_DeployedProcesses_API" repository="5"
isLibrary="false" projectType="ESB" projectVersion="7.2.0-SNAPSHOT"
projectId="e52c9786-f51e-4404-9642-c7727715d418">
        <Properties>
            <Properties>
                <ProjectProperty>
                    <Key>disableWebService</Key>
                    <Value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:boolean">false</Value>
                </ProjectProperty>
            </Properties>
        </Properties>
        <RepositoryPaths>
            <RepositoryPath>
                <repository>5</repository>
                <project>ESB_XVIER-14717_DeployedProcesses_API</project>
                <projectSegments>Adapters</projectSegments>
                <lastProjectSegmentIndex>0</lastProjectSegmentIndex>
                <isDirectory>true</isDirectory>
                <repositoryMembership>INCLUDED</repositoryMembership>
            </RepositoryPath>
            <RepositoryPath>
                <repository>5</repository>
                <project>ESB_XVIER-14717_DeployedProcesses_API</project>
                <projectSegments>ESB_XVIER-14717_DeployedProcesses_API.wsinc</
projectSegments>
                <lastProjectSegmentIndex>0</lastProjectSegmentIndex>
                <isDirectory>false</isDirectory>
                <repositoryMembership>INCLUDED</repositoryMembership>
            </RepositoryPath>
        </RepositoryPaths>
    </Project>
    <Project name="ESBProject_1" repository="5" isLibrary="false" projectType="ESB"
 projectVersion="7.2.0-SNAPSHOT" projectId="6180c7b1-b858-4b7e-9f5f-b11d8768ed68">
        <Properties>
            <Properties>
                <ProjectProperty>
                    <Key>disableWebService</Key>
                    <Value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:boolean">false</Value>
                </ProjectProperty>
            </Properties>
        </Properties>
        <RepositoryPaths>
            <RepositoryPath>
                <repository>5</repository>
                <project>ESBProject_1</project>
                <projectSegments>Adapters</projectSegments>
                <lastProjectSegmentIndex>0</lastProjectSegmentIndex>
```

```
                <isDirectory>true</isDirectory>
                <repositoryMembership>INCLUDED</repositoryMembership>
            </RepositoryPath>
            <RepositoryPath>
                <repository>5</repository>
                <project>ESBProject_1</project>
                <projectSegments>ESBProject_1.wsinc</projectSegments>
                <lastProjectSegmentIndex>0</lastProjectSegmentIndex>
                <isDirectory>false</isDirectory>
                <repositoryMembership>INCLUDED</repositoryMembership>
            </RepositoryPath>
        </RepositoryPaths>
    </Project>
</GetProjectsResponse>
```

10.21.2 Global Parameter Management

`Global Parameter Management`: Verwaltet die globalen Parameter im definierten Parameterspeicher.

Eigenschaften (Properties)

`Operation`	Hier legen Sie fest, welche Operation der Adapter ausführt. **Mögliche Werte:** • `AddKeys`: Fügt alle Parameter in den jeweiligen Parametergruppen hinzu • `Export`: Exportiert alle Parameter aus dem Parameterspeicher (benötigt kein Input-Dokument) • `Get`: Gibt alle Parameter der angefragten Parametergruppen unter Berücksichtigung des Caches aus • `Import`: Alle Parameter in den jeweiligen Parametergruppen werden hinzugefügt oder ersetzt • `Refresh`: Lädt die Parameter aus der Datenbank neu • `RemoveKeys`: Entfernt alle Parameter in den jeweiligen Parametergruppen • `RemoveGroups`: Entfernt alle Parametergruppen • `ReplaceGroups`: Ersetzt alle Parameter in den jeweiligen Parametergruppen

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.plugins.globalparameter.adapter.GlobalParameterStorageAdapter*: Hauptklasse (Standard)

Statuswerte

`1 (successful)`	Die Operation war erfolgreich
`-1 (failed)`	Die Operation schlug aufgrund eines technischen Fehlers fehl

ⓘ Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Input

Für die Operationen *`Get`*, *`AddKeys`*, *`RemoveKeys`*, *`RemoveGroups`*, *`ReplaceGroups`* und *`Import`* werden folgende Input-XML-Dokumente erwartet:

Operation: Get

Folgende Input-XML-Struktur lädt die Gruppen `DEV` und `DEFAULT` aus dem Parameterspeicher:

```
<?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEV"/>
    <Group name="DEFAULT"/>
</Placeholder>
```

Operation: AddKeys & ReplaceGroups & Import

Folgendes XML-Dokument beschreibt Parameter zum Hinzufügen, Ersetzen und Importieren:

```
<?xml version="1.0" encoding="UTF-8"?>
<Placeholder><!-- Bei Operation Import optionales Attribut (default=false):
deleteAndImport=true|false -->
    <Group name="DEV">
        <Key name="message" type="String">Hallo X4 Entwickler!</Key><!--  gültige
Typen: String|Password|Boolean|Integer|Decimal|Date -->
    </Group>
    <Group name="DEFAULT">
        <Key name="message" type="String">Hallo Welt!</Key>
        <Key name="endOfTheWorld" type="Date">2012-12-21</Key>
        <Key name="sayHello" type="Boolean">true</Key>
    </Group>
    <Group name="PASSW">
        <Key name="Password" type="Password">MyNewPassword</Key><!--  gültige Typen:
String|Password|Boolean|Integer|Decimal|Date -->
    </Group>
</Placeholder>
```

> Eine leere Parametergruppe beim Aufruf der Operation `Replace` führt dazu, dass die gesamte Parametergruppe geleert wird. Der Aufruf entspricht somit der Operation `Remove`. Bei der Operation `Import` ist es optional möglich, die entsprechenden Parametergruppen vor dem Import zu Löschen. Hierfür muss im Wurzel-Knoten das Attribut `deleteAndImport` auf *true* gesetzt werden.

Operation: RemoveKeys

Folgendes XML-Dokument entfernt einzelne Parameter einer Gruppe:

```
 <?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEFAULT">
        <Key name="endOfTheWorld"/>
    </Group>
</Placeholder>
```

Operation: RemoveGroups

Folgendes XML-Dokument entfernt eine ganze Parametergruppe:

```
 <?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEFAULT" />
</Placeholder>
```

Ausgabe

Die Operation `Refresh` gibt den aktuellen Status aller Parameter in einem XML-Dokument zurück.

Die Operationen geben die angefragten oder durch die Anfrage geänderten Parametergruppen zurück.

```
<?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEV">
        <Key name="message" type="String">Hallo X4-Entwickler!</Key>
    </Group>
    <Group name="DEFAULT">
        <Key name="message" type="String">Hallo Welt!</Key>
        <Key name="endOfTheWorld" type="Date">2012-12-21</Key>
        <Key name="sayHello" type="Boolean">true</Key>
    </Group>
    <Group name="PASSW">
        <Key name="Password" type="Password">MyNewPassword</Key>
    </Group>
</Placeholder>
```

10.21.3 Keycloak Management

`Keycloak Management`: Verwaltet Benutzer, Gruppen und Rollen in Keycloak.

> **Hinweis**
> Wenn ein anderer **Realm** als der Standard-X4-Realm in **Keycloak** (http://localhost:8085/auth) angelegt wurde, muss dem jeweiligen Client (`http://localhost:8085/auth/admin/master/console/#/YourRealm/clients`) die entsprechende Rolle in **Service Account Roles** zugewiesen werden.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operationen **Mögliche Werte:** • `Activate User`: Benutzer aktivieren • `Create new User`: Neuen Benutzer erstellen • `Deactivate User`: Benutzer deaktivieren • `Delete User`: Benutzer löschen • `Get User by Id`: Benutzerdaten anhand der ID auslesen • `List Users`: Benutzer auflisten • `Update User`: Benutzer aktualisieren • `Add Member to Group`: Benutzer einer Gruppe hinzufügen • `Create new Group`: Gruppe erstellen • `Create new Sub-Group`: Sub-Gruppe erstellen • `Delete Group`: Gruppe löschen • `Get Group by Id`: Gruppendaten anhand der ID auslesen • `Get Members of Group`: Gruppenmitglieder auflisten • `List Groups`: Gruppen auflisten • `Remove Member from Group`: Benutzer aus einer Gruppe entfernen • `Update Group`: Gruppe aktualisieren • `Add Composite to Role`: Untergeordnete Rollen einer zusammengesetzten Rolle hinzufügen • `Add Roles to Group`: Rollen einer Gruppe hinzufügen • `Add Roles to User`: Rollen einem Benutzer hinzufügen • `Create new Role`: Rolle erstellen • `Delete Role`: Rolle löschen • `Get Composite of Role`: Untergeordnete Rollen einer zusammengesetzten Rolle ausgeben • `Get Role by Id`: Rollendaten anhand der ID auslesen • `Get Role by Name`: Rollendaten anhand des Namens auslesen • `Get Users with Role`: Benutzer mit bestimmter Rolle ausgeben • `List Roles`: Rollen auflisten • `Remove Roles From a Group`: Rollen einer Gruppe entfernen • `Remove Roles from a User`: Rollen eines Benutzers entfernen • `Update Role`: Rolle aktualisieren

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.auth.keycloak.KeycloakAdapter*: Hauptklasse (Standard)
`Realm`	Bereich, der in Keycloak REST API-Aufrufen verwendet wird.
`Client ID`	Client ID für die Keycloak REST API-Authentifizierung
`Client Secret`	Geheimer Client-Schlüssel für die Keycloak REST API-Authentifizierung

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters ist ein Fehler aufgetreten.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Weitere Informationen:

- Benutzer
- Gruppen
- Rollen

10.21.3.1 Benutzer

Mit den Operationen können Benutzer (sog. Users) in Keycloak verwaltet werden.

Weitere Informationen finden Sie im Abschnitt

- Create new User
- List Users
- Get User by Id
- Update User
- Activate User
- Deactivate User
- Delete User

10.21.3.1.1 Create new User

Die Operation `Create new User` erstellt in Keycloak einen neuen Benutzer. Die Daten werden in der vom Adapter erwarteten Input-XML definiert.

10.21.3.1.1.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<User>
    <Username></Username>
    <FirstName></FirstName>
    <LastName></LastName>
    <Email></Email>
    <EmailVerified></EmailVerified>
    <Enabled></Enabled>
    <Attributes>
        <Attribute>
            <Name></Name>
            <Values>
                <Value></Value>
            </Values>
        </Attribute>
    </Attributes>
    <Credentials>
        <Credential>
            <Type></Type>
            <Value></Value>
            <Temporary></Temporary>
        </Credential>
    </Credentials>
</User>
```

User

Element	Beschreibung	Mögliche Werte
`User`		Das Element enthält folgende Elemente: • `Username` • `FirstName` (optional) • `LastName` (optional) • `Email` (optional) • `EmailVerified` (optional) • `Enabled` (optional) • `Attributes` (optional) • `Credentials` (optional)
`Username`	Der vom Benutzer gewünschte Benutzername	Beliebiger String
`FirstName`	Der Vorname des Benutzers	Beliebiger String
`LastName`	Der Nachname des Benutzers	Beliebiger String
`Email`	Die E-Mail-Adresse des Benutzers	E-Mail-Adresse
`EmailVerified`	Angabe, ob die angegebene E-Mail-Adresse verifiziert ist	• `true`: Die E-Mail-Adresse ist verifiziert. • `false`: Die E-Mail-Adresse ist nicht verifiziert.

Element	Beschreibung	Mögliche Werte
`Enabled`	Angabe, ob der Benutzer aktiviert oder deaktiviert ist	• `true`: Der Benutzer ist aktiviert. • `false`: Der Benutzer ist deaktiviert.
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)
`Credentials`	Benutzeranmeldeinformationen	Das Element enthält folgendes Element: • `Credential`

Attributes

Element	Beschreibung	
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)
`Attribute`	Personalisiertes, konfigurierbares Attribut	Das Element enthält folgende Elemente: • `Name` • `Values`
`Name`	Attributname	Beliebiger String
`Values`	Attributwerte	Das Element enthält folgende Elemente: • `Value` (Mehrfachdeklaration möglich)
`Value`	Attributwert	Beliebiger String

Credentials

Im `Credentials`-Element wird die Art des Anmeldedatensatzes und der dazugehörige Wert festgelegt. Das kann zum Beispiel ein Passwort sein.

Element	Beschreibung	Mögliche Werte
`Credentials`	Benutzeranmeldeinformationen	Das Element enthält folgende Elemente: • `Credential`

Element	Beschreibung	Mögliche Werte
`Credential`	Anmeldedatensatz	Das Element enthält folgende Elemente: • `Type` • `Value` • `Temporary`
`Type`	Art des Anmeldedatensatzes, z. B. Passwort	`password`: Der Anmeldedatensatz ist ein Passwort.
`Value`	Attributwert	Beliebiger String
`Temporary`	Angabe, ob das Passwort temporär ist	`true`: Das Passwort ist temporär. `false`: Das Passwort ist nicht temporär.

XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="User">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Username"/>
                <xs:element ref="FirstName"/>
                <xs:element ref="LastName"/>
                <xs:element ref="Email"/>
                <xs:element ref="EmailVerified"/>
                <xs:element ref="Enabled"/>
                <xs:element ref="Attributes"/>
                <xs:element ref="Credentials"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Username" type="xs:NCName"/>
    <xs:element name="FirstName" type="xs:NCName"/>
    <xs:element name="LastName" type="xs:NCName"/>
    <xs:element name="Email" type="xs:string"/>
    <xs:element name="EmailVerified" type="xs:boolean"/>
    <xs:element name="Enabled" type="xs:boolean"/>
    <xs:element name="Attributes">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Attribute"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Attribute">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Name"/>
                <xs:element ref="Values"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Values">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Value"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Credentials">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Credential"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Credential">
        <xs:complexType>
```

```
            <xs:sequence>
                <xs:element ref="Type"/>
                <xs:element ref="Value"/>
                <xs:element ref="Temporary"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Type" type="xs:NCName"/>
    <xs:element name="Temporary" type="xs:boolean"/>
    <xs:element name="Value" type="xs:string"/>
</xs:schema>
```

10.21.3.1.1.2 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<User>
    <Username>JohnDoe</Username>
    <FirstName>John</FirstName>
    <LastName>Doe</LastName>
    <Email>John.Doe@example.com</Email>
    <EmailVerified>true</EmailVerified>
    <Enabled>true</Enabled>
    <Attributes>
        <Attribute>
            <Name>Employment Relationship</Name>
            <Values>
                <Value>Software Developer</Value>
                <Value>Sub-Team Lead</Value>
            </Values>
        </Attribute>
    </Attributes>
    <Credentials>
        <Credential>
            <Type>password</Type>
            <Value>password123</Value>
            <Temporary>false</Temporary>
        </Credential>
    </Credentials>
</User>
```

10.21.3.1.2 List Users

Die Operation `List Users` listet alle Benutzer auf, die in Keycloak existieren.

Input-Struktur

Der Adapter erwartet keinen Input. Ohne Input werden aber nur die ersten 100 Benutzer zurückgegeben. Um mehr Benutzer aufzulisten, verwenden Sie folgenden Input:

```
<KeycloakListUsersParameters>
<briefRepresentation></briefRepresentation>
<firstName></firstName>
<lastName></lastName>
<username>demo_1</username>
<email></email>
<offset></offset>
<limit>500</limit>
</KeycloakListUsersParameters>
```

Die Funktion der Parameter `briefRepresentation`, `firstName`, `lastName`, `username` und `email` entspricht der Funktion in der X4 API (Swagger UI), die Sie über das X4 Control Center aufrufen können. Sie können diese Parameter im Input-Dokument leer lassen.

Die Parameter `offset` und `limit` haben folgende Funktion:

- `offset`: Anzahl der Benutzer, die ab dem Anfang der Benutzerliste übersprungen werden.
- `limit`: Maximale Anzahl der Benutzer, die zurückgegeben werden soll. Die angegebene Anzahl an Benutzern wird unabhängig davon zurückgegeben, wie viele Benutzer mit dem Parameter `offset` übersprungen werden. Beispiel: Wenn bei 600 Benutzern `offset` auf 50 und `limit` auf 500 gesetzt wird, wird eine Liste mit den Benutzern 51–550 zurückgegeben.

10.21.3.1.2.1 Beispiele

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Users>
    <User>
        <Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
        <CreatedTimestamp>1634133393343</CreatedTimestamp>
        <Username>exampleuser</Username>
        <Enabled>true</Enabled>
        <Totp>false</Totp>
        <EmailVerified>true</EmailVerified>
        <FirstName>Jane</FirstName>
        <LastName>Doe</LastName>
        <Email>john.doe@example.com</Email>
        <Attributes>
            <Attribute>
                <Name>Employment Relationship</Name>
                <Values>
                    <Value>Software Developer</Value>
                    <Value>Sub-Team Lead</Value>
                </Values>
            </Attribute>
        </Attributes>
        <NotBefore>0</NotBefore>
        <access>
            <entry>
                <key>manageGroupMembership</key>
                <value>true</value>
            </entry>
            <entry>
                <key>view</key>
                <value>true</value>
            </entry>
            <entry>
                <key>mapRoles</key>
                <value>true</value>
            </entry>
            <entry>
                <key>impersonate</key>
                <value>false</value>
            </entry>
            <entry>
                <key>manage</key>
                <value>true</value>
            </entry>
        </access>
    </User>
</Users>
```

10.21.3.1.3 Get User by Id

Die Operation `Get User by Id` gibt die Daten des Benutzers mit der angegebenen ID aus.

Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

10.21.3.1.3.1 Id

Element	Beschreibung	Mögliche Werte
Id	UUID des Benutzers	UUID

10.21.3.1.3.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Id" type="xs:string"/>
</xs:schema>
```

10.21.3.1.3.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<User>
    <Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
    <CreatedTimestamp>1634133393343</CreatedTimestamp>
    <Username>exampleuser</Username>
    <Enabled>true</Enabled>
    <Totp>false</Totp>
    <EmailVerified>true</EmailVerified>
    <FirstName>Jane</FirstName>
    <LastName>Doe</LastName>
    <Email>john.doe@example.com</Email>
    <Attributes>
        <Attribute>
            <Name>Employment Relationship</Name>
            <Values>
                <Value>Software Developer</Value>
                <Value>Sub-Team Lead</Value>
            </Values>
        </Attribute>
    </Attributes>
    <NotBefore>0</NotBefore>
    <access>
        <entry>
            <key>manageGroupMembership</key>
            <value>true</value>
        </entry>
        <entry>
            <key>view</key>
            <value>true</value>
        </entry>
        <entry>
            <key>mapRoles</key>
            <value>true</value>
        </entry>
        <entry>
            <key>impersonate</key>
            <value>false</value>
        </entry>
        <entry>
            <key>manage</key>
            <value>true</value>
        </entry>
    </access>
</User>
```

10.21.3.1.4 Update User

Die Operation `Update User` aktualisiert die Daten des Benutzers mit der angegebenen ID. Die Daten werden in der vom Adapter erwarteten Input-XML definiert.

> ⚠ Der Benutzername kann nicht aktualisiert werden.

10.21.3.1.4.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<User>
    <Id></Id>
    <Enabled></Enabled>
    <Totp></Totp>
    <EmailVerified></EmailVerified>
    <FirstName></FirstName>
    <LastName></LastName>
    <Email></Email>
    <Attributes>
        <Attribute>
            <Name></Name>
            <Values>
                <Value></Value>
            </Values>
        </Attribute>
    </Attributes>
    <Credentials>
        <Credential>
            <Type></Type>
            <Value></Value>
            <Temporary></Temporary>
        </Credential>
    </Credentials>
    <RequiredActions></RequiredActions>
    <NotBefore></NotBefore>
</User>
```

Element	Beschreibung	Mögliche Werte
`User`	Benutzerdefinition	Das Element enthält folgende Elemente: • `Id` • `FirstName` (optional) • `LastName` (optional) • `Email` (optional) • `EmailVerified` (optional) • `Enabled` (optional) • `Totp` (optional) • `Attributes` (optional) • `Credentials` (optional) • `RequiredActions` (optional) • `NotBefore` (optional)
`Id`	Eindeutige ID des Benutzers	UUID
`FirstName`	Der Vorname des Benutzers	Beliebiger String
`LastName`	Der Nachname des Benutzers	Beliebiger String
`Email`	Die E-Mail-Adresse des Benutzers	E-Mail-Adresse

Element	Beschreibung	Mögliche Werte
`EmailVerified`	Angabe, ob die angegebene E-Mail-Adresse verifiziert ist	• `true`: Die E-Mail-Adresse ist verifiziert. • `false`: Die E-Mail-Adresse ist nicht verifiziert.
`Enabled`	Angabe, ob der Benutzer aktiviert oder deaktiviert ist	• `true`: Der Benutzer ist aktiviert. • `false`: Der Benutzer ist deaktiviert.
`Totp`	Angabe, ob das Passwort nach der ersten Anmeldung ungültig ist	• `true`: Der Benutzer muss nach der ersten Anmeldung das Passwort ändern. • `false`: Der Benutzer muss das Passwort nicht ändern.
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)
`Credentials`	Benutzeranmeldeinformationen	Das Element enthält folgendes Element: • `Credential`
`RequiredActions`	Angabe, ob vom Benutzer zur Handlung aufgefordert wird, sobald der Benutzer sich anmeldet.	`VERIFY_EMAIL`: Der Benutzer muss nach der Anmeldung seine E-Mail-Adresse verifizieren. `UPDATE_PROFILE`: Der Benutzer muss nach der Anmeldung sein Benutzerprofil aktualisieren. `UPDATE_PASSWORD`: Der Benutzer muss nach der Anmeldung sein Passwort aktualisieren.
`NotBefore`	Die Anzahl der fehlgeschlagenen Anmeldeversuche	Ganzzahl

Attributes

Element	Beschreibung	
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)

Element	Beschreibung	
`Attribute`	Personalisiertes, konfigurierbares Attribut	Das Element enthält folgende Elemente: • `Name` • `Values`
`Name`	Attributname	Beliebiger String
`Values`	Attributwerte	Das Element enthält folgende Elemente: • `Value` (Mehrfachdeklaration möglich)
`Value`	Attributwert	Beliebiger String

Credentials

Im `Credentials`-Element wird die Art des Anmeldedatensatzes und der dazugehörige Wert festgelegt. Das kann zum Beispiel ein Passwort sein.

Element	Beschreibung	Mögliche Werte
`Credentials`	Benutzeranmeldeinformationen	Das Element enthält folgende Elemente: • `Credential`
`Credential`	Anmeldedatensatz	Das Element enthält folgende Elemente: • `Type` • `Value` • `Temporary`
`Type`	Art des Anmeldedatensatzes, z. B. Passwort	`password`: Der Anmeldedatensatz ist ein Passwort.
`Value`	Attributwert	Beliebiger String
`Temporary`	Angabe, ob das Passwort temporär ist	`true`: Das Passwort ist temporär. `false`: Das Passwort ist nicht temporär.

XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="User">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Id"/>
                <xs:element ref="Enabled"/>
                <xs:element ref="Totp"/>
                <xs:element ref="EmailVerified"/>
                <xs:element ref="FirstName"/>
                <xs:element ref="LastName"/>
                <xs:element ref="Email"/>
                <xs:element ref="Attributes"/>
                <xs:element ref="Credentials"/>
                <xs:element ref="RequiredActions"/>
                <xs:element ref="NotBefore"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Id" type="xs:string"/>
    <xs:element name="Enabled" type="xs:boolean"/>
    <xs:element name="Totp" type="xs:boolean"/>
    <xs:element name="EmailVerified" type="xs:boolean"/>
    <xs:element name="FirstName" type="xs:NCName"/>
    <xs:element name="LastName" type="xs:NCName"/>
    <xs:element name="Email" type="xs:string"/>
    <xs:element name="Attributes">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Attribute"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Attribute">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Name"/>
                <xs:element ref="Values"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Values">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Value"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Credentials">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Credential"/>
            </xs:sequence>
```

```
        </xs:complexType>
    </xs:element>
    <xs:element name="Credential">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Type"/>
                <xs:element ref="Value"/>
                <xs:element ref="Temporary"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Type" type="xs:NCName"/>
    <xs:element name="Temporary" type="xs:boolean"/>
    <xs:element name="RequiredActions" type="xs:NCName"/>
    <xs:element name="NotBefore" type="xs:integer"/>
    <xs:element name="Value" type="xs:string"/>
</xs:schema>
```

10.21.3.1.4.2 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<User>
    <Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
    <Enabled>true</Enabled>
    <Totp>false</Totp>
    <EmailVerified>true</EmailVerified>
    <FirstName>Jane</FirstName>
    <LastName>Doe</LastName>
    <Email>john.doe@example.com</Email>
    <Attributes>
        <Attribute>
            <Name>Employment Relationship</Name>
            <Values>
                <Value>Software Developer</Value>
                <Value>Sub-Team Lead</Value>
            </Values>
        </Attribute>
    </Attributes>
    <Credentials>
        <Credential>
            <Type>password</Type>
            <Value>password123</Value>
            <Temporary>false</Temporary>
        </Credential>
    </Credentials>
    <RequiredActions>VERIFY_EMAIL</RequiredActions>
    <NotBefore>0</NotBefore>
</User>
```

10.21.3.1.5 Activate User

Die Operation `Activate User` aktiviert den Benutzer mit der angegebenen ID.

> ⚠ Der Benutzer muss in Keycloak existieren.

10.21.3.1.5.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Beschreibung	Mögliche Werte
`Id`	UUID des Benutzers	UUID

10.21.3.1.5.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Id" type="xs:string"/>
</xs:schema>
```

10.21.3.1.5.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
```

10.21.3.1.6 Deactivate User

Die Operation `Deactivate User` deaktiviert den Benutzer mit der angegebenen ID.

> ⚠ Der Benutzer muss in Keycloak existieren.

10.21.3.1.6.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Beschreibung	Mögliche Werte
`Id`	UUID des Benutzers	UUID

10.21.3.1.6.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Id" type="xs:string"/>
</xs:schema>
```

10.21.3.1.6.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
```

10.21.3.1.7 Delete User

Die Operation `Delete User` löscht den Benutzer mit der angegebenen ID.

> ⚠ Der Benutzer muss in Keycloak vorhanden sein.
> Ein gelöschter Benutzer kann nicht wiederhergestellt werden.

10.21.3.1.7.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Beschreibung	Mögliche Werte
`Id`	UUID des Benutzers	UUID

10.21.3.1.7.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Id" type="xs:string"/>
</xs:schema>
```

10.21.3.1.7.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
```

10.21.3.2 Gruppen

Mit den Operationen können Gruppen (sog. Groups) in Keycloak verwaltet werden.

Weitere Informationen finden Sie im Abschnitt

- Create new Group
- Create new Sub-Group
- List Groups
- Get Group by Id
- Update Group
- Delete Group
- Add Member to Group
- Get Members of Group
- Remove Member from Group

10.21.3.2.1 Create new Group

Die Operation `Create new Group` erstellt in Keycloak eine neue Gruppe. Die Daten werden in der vom Adapter erwarteten Input-XML definiert.

10.21.3.2.1.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Group>
    <Name></Name>
    <Attributes>
        <Attribute>
            <Name></Name>
            <Values>
                <Value></Value>
            </Values>
        </Attribute>
    </Attributes>
</Group>
```

Group

Element	Beschreibung	Mögliche Werte
`Group`	Gruppendefinition	Das Element enthält folgende Elemente: • `Name` • `Attributes` (optional)

Element	Beschreibung	Mögliche Werte
`Name`	Name der Gruppe	Beliebiger String
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)

Attributes

Element	Beschreibung	
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)
`Attribute`	Personalisiertes, konfigurierbares Attribut	Das Element enthält folgende Elemente: • `Name` • `Values`
`Name`	Attributname	Beliebiger String
`Values`	Attributwerte	Das Element enthält folgende Elemente: • `Value` (Mehrfachdeklaration möglich)
`Value`	Attributwert	Beliebiger String

10.21.3.2.1.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Group">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Id"/>
                <xs:element ref="Name"/>
                <xs:element ref="Path"/>
                <xs:element ref="Attributes"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Id" type="xs:string"/>
    <xs:element name="Path" type="xs:string"/>
    <xs:element name="Attributes">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Attribute"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Attribute">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Name"/>
                <xs:element ref="Values"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Values">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Value"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Value" type="xs:NCName"/>
    <xs:element name="Name" type="xs:NCName"/>
</xs:schema>
```

10.21.3.2.1.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Group>
    <Name>Customer</Name>
    <Attributes>
        <Attribute>
            <Name>Company</Name>
            <Values>
                <Value>SoftProject</Value>
            </Values>
        </Attribute>
    </Attributes>
</Group>
```

10.21.3.2.2 Create new Sub-Group

Die Operation `Create new Group` erstellt in Keycloak eine neue Untergruppe. Die Daten werden in der vom Adapter erwarteten Input-XML definiert.

10.21.3.2.2.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<ParentGroup>
    <ParentId></ParentId>
    <SubGroup>
        <Name></Name>
        <Attributes>
            <Attribute>
                <Name></Name>
                <Values>
                    <Value></Value>
                </Values>
            </Attribute>
        </Attributes>
    </SubGroup>
</ParentGroup>
```

ParentGroup

Element	Beschreibung	Mögliche Werte
`ParentGroup`		Das Element enthält folgende Elemente: • `ParentId` • `SubGroup`
`ParentId`	UUID der übergeordneten Gruppe	UUID
`SubGroup`	Untergruppendefinition	Das Element enthält folgende Elemente: • `Name` • `Attributes` (optional)

SubGroup

Element	Beschreibung	Mögliche Werte
`SubGroup`	Untergruppendefinition	Das Element enthält folgende Elemente: • `Name` • `Attributes` (optional)
`Name`	Name der Untergruppe	Beliebiger String
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)

Attributes

Element	Beschreibung	
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)
`Attribute`	Personalisiertes, konfigurierbares Attribut	Das Element enthält folgende Elemente: • `Name` • `Values`
`Name`	Attributname	Beliebiger String
`Values`	Attributwerte	Das Element enthält folgende Elemente: • `Value` (Mehrfachdeklaration möglich)
`Value`	Attributwert	Beliebiger String

10.21.3.2.2.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="ParentGroup">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="ParentId"/>
                <xs:element ref="SubGroup"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ParentId" type="xs:string"/>
    <xs:element name="SubGroup">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Name"/>
                <xs:element ref="Attributes"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Attributes">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Attribute"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Attribute">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Name"/>
                <xs:element ref="Values"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Values">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Value"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Value" type="xs:string"/>
    <xs:element name="Name" type="xs:string"/>
</xs:schema>
```

10.21.3.2.2.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<ParentGroup>
    <ParentId>49bf25f9-7878-484e-87da-bebff1550379</ParentId>
    <SubGroup>
        <Name>Developer</Name>
        <Attributes>
            <Attribute>
                <Name>Team</Name>
                <Values>
                    <Value>Red</Value>
                </Values>
            </Attribute>
        </Attributes>
    </SubGroup>
</ParentGroup>
```

10.21.3.2.3 List Groups

Die Operation `List Groups` listet alle Gruppen auf, die in Keycloak existieren.

10.21.3.2.3.1 Beispiele

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Groups>
    <Group>
        <Id>e8863a97-771d-4c34-90e8-4698d06928a5</Id>
        <Name>Employee</Name>
        <Path>/Employee</Path>
        <Attributes/>
        <ClientRoles/>
        <SubGroups>
            <Group>
                <Id>21181305-1361-469a-845e-873f56025304</Id>
                <Name>Developer</Name>
                <Path>/Employee/Developer</Path>
                <Attributes/>
                <ClientRoles/>
                <SubGroups/>
            </Group>
            <Group>
                <Id>531b9720-79db-4ef9-9f81-a026b6ebe535</Id>
                <Name>Consultant</Name>
                <Path>/Employee/Consultant</Path>
                <Attributes/>
                <ClientRoles/>
                <SubGroups/>
            </Group>
        </SubGroups>
    </Group>
    <Group>
        <Id>78f84034-3129-4636-97c5-189b338deeb4</Id>
        <Name>Customer</Name>
        <Path>/Customer</Path>
        <Attributes/>
        <ClientRoles/>
        <SubGroups/>
    </Group>
</Groups>
```

10.21.3.2.4 Get Group by Id

Die Operation `Get Group by Id` gibt die Daten der Gruppe mit der angegebenen ID aus.

10.21.3.2.4.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Beschreibung	Mögliche Werte
`Id`	UUID der Gruppe	UUID

10.21.3.2.4.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Id" type="xs:string"/>
</xs:schema>
```

10.21.3.2.4.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>78f84034-3129-4636-97c5-189b338deeb4</Id>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Group>
    <Id>78f84034-3129-4636-97c5-189b338deeb4</Id>
    <Name>Employee</Name>
    <Path>/Employee</Path>
    <Attributes>
         <Attribute>
            <Name>Employment Relationship</Name>
            <Values>
                <Value>Software Developer</Value>
            </Values>
        </Attribute>
    </Attributes>
    <ClientRoles/>
    <SubGroups/>
    <access>
        <entry>
            <key>view</key>
            <value>true</value>
        </entry>
        <entry>
            <key>manage</key>
            <value>true</value>
        </entry>
        <entry>
            <key>manageMembership</key>
            <value>true</value>
        </entry>
    </access>
</Group>
```

10.21.3.2.5 Update Group

Die Operation `Update Group` aktualisiert die Daten der Gruppe mit der angegebenen ID. Die Daten werden in der vom Adapter erwarteten Input-XML definiert.

10.21.3.2.5.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Group>
    <Id></Id>
    <Name></Name>
    <Attributes>
        <Attribute>
            <Name></Name>
            <Values>
                <Value></Value>
            </Values>
        </Attribute>
    </Attributes>
</Group>
```

Group

Element	Beschreibung	Mögliche Werte
`Group`	Gruppendefinition	Das Element enthält folgende Elemente: • `Id` • `Name` • `Attributes` (optional)
`Id`	UUID der Gruppe	UUID
`Name`	Name der Gruppe	Beliebiger String
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)

Attributes

Element	Beschreibung	
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)

Element	Beschreibung	
`Attribute`	Personalisiertes, konfigurierbares Attribut	Das Element enthält folgende Elemente: • `Name` • `Values`
`Name`	Attributname	Beliebiger String
`Values`	Attributwerte	Das Element enthält folgende Elemente: • `Value` (Mehrfachdeklaration möglich)
`Value`	Attributwert	Beliebiger String

10.21.3.2.5.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Group">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Id"/>
                <xs:element ref="Name"/>
                <xs:element ref="Path"/>
                <xs:element ref="Attributes"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Id" type="xs:string"/>
    <xs:element name="Path" type="xs:string"/>
    <xs:element name="Attributes">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Attribute"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Attribute">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Name"/>
                <xs:element ref="Values"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Values">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Value"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Value" type="xs:string"/>
    <xs:element name="Name" type="xs:string"/>
</xs:schema>
```

10.21.3.2.5.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Group>
    <Id>e8863a97-771d-4c34-90e8-4698d06928a5</Id>
    <Name>Customer</Name>
    <Path>/Customer</Path>
    <Attributes>
        <Attribute>
            <Name>Company</Name>
            <Values>
                <Value>SoftProject</Value>
            </Values>
        </Attribute>
    </Attributes>
</Group>
```

10.21.3.2.6 Delete Group

Die Operation `Delete User` löscht den Benutzer mit der angegebenen ID.

> ⚠ Die Gruppe muss in Keycloak existieren.
> Eine gelöschte Gruppe kann nicht wiederhergestellt werden.

10.21.3.2.6.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Beschreibung	Mögliche Werte
`Id`	UUID der Gruppe	UUID

10.21.3.2.6.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Id" type="xs:string"/>
</xs:schema>
```

10.21.3.2.6.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
```

10.21.3.2.7 Add Member to Group

Die Operation `Add Member to Group` fügt einen Benutzer einer Gruppe hinzu. Die Daten werden in der vom Adapter erwarteten Input-XML definiert.

10.21.3.2.7.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Ids>
    <UserId></UserId>
    <GroupId></GroupId>
</Ids>
```

Ids

Element	Beschreibung	Mögliche Werte
`Ids`		Das Element enthält folgende Elemente: • `UserId` • `GroupId`
`UserId`	UUID des Benutzers	UUID
`GroupId`	UUID der Gruppe	UUID

10.21.3.2.7.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Ids">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="UserId"/>
                <xs:element ref="GroupId"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="UserId" type="xs:string"/>
    <xs:element name="GroupId" type="xs:string"/>
</xs:schema>
```

10.21.3.2.7.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Ids>
    <UserId>f37db159-8c3c-4b0a-ae7f-5362dcf4ed1b</UserId>
    <GroupId>8445918e-f3ba-4c93-b3c4-7afd7b65fa57</GroupId>
</Ids>
```

10.21.3.2.8 Get Members of Group

Die Operation `Get Members of Group` gibt die Mitglieder der Gruppe mit der angegebenen ID aus.

10.21.3.2.8.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Beschreibung	Mögliche Werte
`Id`	UUID der Gruppe	UUID

10.21.3.2.8.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Id" type="xs:string"/>
</xs:schema>
```

10.21.3.2.8.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>e8863a97-771d-4c34-90e8-4698d06928a5</Id>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Users>
    <User>
        <Id>a7bbfc23-a129-47c6-a802-e748d73876a6</Id>
        <CreatedTimestamp>1634287894774</CreatedTimestamp>
        <Username>jane</Username>
        <Enabled>true</Enabled>
        <Totp>false</Totp>
        <EmailVerified>true</EmailVerified>
        <FirstName>Jane</FirstName>
        <LastName>Doe</LastName>
        <Email>jane@example.com</Email>
        <NotBefore>0</NotBefore>
    </User>
    <User>
        <Id>1e6d7aed-8d82-4926-bf8d-fc490002f377</Id>
        <CreatedTimestamp>1634287963141</CreatedTimestamp>
        <Username>john</Username>
        <Enabled>true</Enabled>
        <Totp>false</Totp>
        <EmailVerified>true</EmailVerified>
        <FirstName>John</FirstName>
        <LastName>Doe</LastName>
        <Email>john@example.com</Email>
        <NotBefore>0</NotBefore>
    </User>
</Users>
```

10.21.3.2.9 Remove Member from Group

Die Operation `Remove Member from Group` entfernt einen Benutzer von einer Gruppe. Die Daten werden in der vom Adapter erwarteten Input-XML definiert.

10.21.3.2.9.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Ids>
    <UserId></UserId>
    <GroupId></GroupId>
</Ids>
```

Ids

Element	Beschreibung	Mögliche Werte
`Ids`		Das Element enthält folgende Elemente: • `UserId` • `GroupId`
`UserId`	UUID des Benutzers	UUID
`GroupId`	UUID der Gruppe	UUID

10.21.3.2.9.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="Ids">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="UserId"/>
        <xs:element ref="GroupId"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="UserId" type="xs:string"/>
  <xs:element name="GroupId" type="xs:string"/>
</xs:schema>
```

10.21.3.2.9.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Ids>
    <UserId>f37db159-8c3c-4b0a-ae7f-5362dcf4ed1b</UserId>
    <GroupId>8445918e-f3ba-4c93-b3c4-7afd7b65fa57</GroupId>
</Ids>
```

10.21.3.3 Rollen

Mit den Operationen können Rollen (sog. Roles) in Keycloak verwaltet werden.

Weitere Informationen finden Sie im Abschnitt

- Create new Role
- List Roles
- Get Role by Id
- Get Role by Name
- Update Role
- Delete Role
- Add Composite to Role
- Get Composite of Role
- Add Roles to User
- Get Users with Role
- Remove Roles from a User
- Add Roles to Group
- Remove Roles From a Group

10.21.3.3.1 Create new Role

Die Operation `Create new Role` erstellt in Keycloak eine neue Rolle. Die Daten werden in der vom Adapter erwarteten Input-XML definiert.

10.21.3.3.1.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Role>
    <Name></Name>
    <Description></Description>
    <Composite></Composite>
    <ContainerId></ContainerId>
    <Attributes>
        <Attribute>
            <Name></Name>
            <Values>
                <Value></Value>
            </Values>
        </Attribute>
    </Attributes>
</Role>
```

Role

Element	Beschreibung	Mögliche Werte
`Role`	Rollendefinition	Das Element enthält folgende Elemente: • `Name` • `Description` optional • `Composite` optional • `ContainerId` • `Attributes` (optional)
`Name`	Name der Rolle	Beliebiger String
`Description`	Beschreibung der Rolle	Beliebiger String
`Composite`	Angabe, ob die Rolle eine zusammengesetzte Rolle ist	• `true`: Die Rolle ist eine zusammengesetzte Rolle. • `false`: Die Rolle ist keine zusammengesetzte Rolle. (Standard)
`ContainerId`	Name des Realm	Beliebiger String
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)

Attributes

Element	Beschreibung	
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)
`Attribute`	Personalisiertes, konfigurierbares Attribut	Das Element enthält folgende Elemente: • `Name` • `Values`
`Name`	Attributname	Beliebiger String
`Values`	Attributwerte	Das Element enthält folgende Elemente: • `Value` (Mehrfachdeklaration möglich)
`Value`	Attributwert	Beliebiger String

10.21.3.3.1.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="Role">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Name"/>
        <xs:element ref="Description"/>
        <xs:element ref="Composite"/>
        <xs:element ref="ContainerId"/>
        <xs:element ref="Attributes"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Description" type="xs:string"/>
  <xs:element name="Composite" type="xs:boolean"/>
  <xs:element name="ContainerId" type="xs:NCName"/>
  <xs:element name="Attributes">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="Attribute"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Attribute">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Name"/>
        <xs:element ref="Values"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Values">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="Value"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Value" type="xs:string"/>
  <xs:element name="Name" type="xs:string"/>
</xs:schema>
```

10.21.3.3.1.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Role>
    <Name>Developer</Name>
    <Description>Software Developer</Description>
    <Composite>false</Composite>
    <ContainerId>exampleRealm</ContainerId>
    <Attributes>
        <Attribute>
            <Name>Team</Name>
            <Values>
                <Value>Blue</Value>
                <Value>Red</Value>
            </Values>
        </Attribute>
    </Attributes>
</Role>
```

10.21.3.3.2 List Roles

Die Operation `List Roles` listet alle Rollen auf, die in Keycloak existieren.

10.21.3.3.2.1 Beispiele

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Roles>
    <Role>
        <Id>6342e31d-64f7-4fee-9441-c2be8f5db579</Id>
        <Name>x4_admin_access</Name>
        <Composite>false</Composite>
        <ClientRole>false</ClientRole>
        <ContainerId>X4Realm</ContainerId>
        <Attributes/>
    </Role>
    <Role>
        <Id>3915229f-7544-4701-b1dc-6092861d9101</Id>
        <Name>x4_dev_access</Name>
        <Composite>false</Composite>
        <ClientRole>false</ClientRole>
        <ContainerId>X4Realm</ContainerId>
        <Attributes/>
    </Role>
</Roles>
```

10.21.3.3.3 Get Role by Id

Die Operation `Get Role by Id` gibt die Daten der Rolle mit der angegebenen ID aus.

10.21.3.3.3.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Beschreibung	Mögliche Werte
Id	UUID der Rolle	UUID

10.21.3.3.3.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Id" type="xs:string"/>
</xs:schema>
```

10.21.3.3.3.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>3915229f-7544-4701-b1dc-6092861d9101</Id>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Role>
    <Id>3915229f-7544-4701-b1dc-6092861d9101</Id>
    <Name>x4_dev_access</Name>
    <Composite>false</Composite>
    <ClientRole>false</ClientRole>
    <ContainerId>X4Realm</ContainerId>
    <Attributes/>
</Role>
```

10.21.3.3.4 Get Role by Name

Die Operation `Get Role by Name` gibt die Daten der Rolle mit dem angegebenen Namen aus.

10.21.3.3.4.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Name></Name>
```

Name

Element	Beschreibung	Mögliche Werte
`Name`	Name der Rolle	Beliebiger String

10.21.3.3.4.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Name" type="xs:string"/>
</xs:schema>
```

10.21.3.3.4.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Name>x4_dev_access</Name>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Role>
    <Id>3915229f-7544-4701-b1dc-6092861d9101</Id>
    <Name>x4_dev_access</Name>
    <Composite>false</Composite>
    <ClientRole>false</ClientRole>
    <ContainerId>X4Realm</ContainerId>
    <Attributes/>
</Role>
```

10.21.3.3.5 Update Role

Die Operation `Update Role` aktualisiert die Daten der Rolle mit der angegebenen ID. Die Daten werden in der vom Adapter erwarteten Input-XML definiert.

10.21.3.3.5.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Role>
    <Id></Id>
    <Name></Name>
    <Description></Description>
    <Composite></Composite>
    <ClientRole></ClientRole>
    <ContainerId></ContainerId>
    <Attributes>
        <Attribute>
            <Name></Name>
            <Values>
                <Value></Value>
            </Values>
        </Attribute>
    </Attributes>
</Role>
```

Role

Element	Beschreibung	Mögliche Werte
`Role`	Rollendefinition	Das Element enthält folgende Elemente: • `Id` • `Name` • `Description` (optional) • `Composite` (optional) • `ClientRole` (optional) • `ContainerId` (optional) • `Attributes` (optional)
`Id`	UUID der Rolle	UUID
`Name`	Name der Rolle	Beliebiger String
`Description`	Beschreibung der Rolle	Beliebiger String
`Composite`	Angabe, ob die Rolle eine zusammengesetzte Rolle ist	• `true`: Die Rolle ist eine zusammengesetzte Rolle. • `false`: Die Rolle ist keine zusammengesetzte Rolle. (Standard)
`ClientRole`	Angabe, ob die Rolle eine Client-Rolle ist. Dabei handelt es sich im Grunde um einen für den Client bestimmten Rollennamensraum.	Beliebiger String
`ContainerId`	Name des Realm	Beliebiger String

Element	Beschreibung	Mögliche Werte
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)

Attributes

Element	Beschreibung	
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)
`Attribute`	Personalisiertes, konfigurierbares Attribut	Das Element enthält folgende Elemente: • `Name` • `Values`
`Name`	Attributname	Beliebiger String
`Values`	Attributwerte	Das Element enthält folgende Elemente: • `Value` (Mehrfachdeklaration möglich)
`Value`	Attributwert	Beliebiger String

10.21.3.3.5.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="Role">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Id"/>
        <xs:element ref="Name"/>
        <xs:element ref="Description"/>
        <xs:element ref="Composite"/>
        <xs:element ref="ClientRole"/>
        <xs:element ref="ContainerId"/>
        <xs:element ref="Attributes"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Id" type="xs:string"/>
  <xs:element name="Description" type="xs:string"/>
  <xs:element name="Composite" type="xs:boolean"/>
  <xs:element name="ClientRole" type="xs:boolean"/>
  <xs:element name="ContainerId" type="xs:NCName"/>
  <xs:element name="Attributes">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="Attribute"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Attribute">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Name"/>
        <xs:element ref="Values"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Values">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="Value"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Value" type="xs:string"/>
  <xs:element name="Name" type="xs:string"/>
</xs:schema>
```

10.21.3.3.5.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Role>
    <Id>658242d5-0caf-4ecd-b930-45c02ccf39d4</Id>
    <Name>Developer</Name>
    <Description>Software Developer</Description>
    <Composite>false</Composite>
    <ClientRole>false</ClientRole>
    <ContainerId>X4Realm</ContainerId>
    <Attributes>
        <Attribute>
            <Name>Team</Name>
            <Values>
                <Value>Blue</Value>
                <Value>Red</Value>
            </Values>
        </Attribute>
    </Attributes>
</Role>
```

10.21.3.3.6 Delete Role

Die Operation `Delete Role` löscht die Rolle mit der angegebenen ID.

> ⚠ Eine gelöschte Rolle kann nicht wiederhergestellt werden.

10.21.3.3.6.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Beschreibung	Mögliche Werte
`Id`	UUID der Rolle	UUID

10.21.3.3.6.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Id" type="xs:string"/>
</xs:schema>
```

10.21.3.3.6.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
```

10.21.3.3.7 Add Composite to Role

Die Operation `Add Composite to Role` fügt einer zusammengesetzten Rolle eine untergeordnete Rolle hinzu. Sobald einem Benutzer die zusammengesetzte Rolle zugewiesen wird, werden dem Benutzer alle in der zusammengesetzten Rolle untergeordneten Rollen automatisch zugewiesen.

10.21.3.3.7.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<ParentRole>
    <ParentId></ParentId>
    <SubRoles>
        <SubRole>
            <Id></Id>
            <Name></Name>
            <Composite></Composite>
            <ContainerId></ContainerId>
            <Attributes>
                <Attribute>
                    <Name></Name>
                    <Values>
                        <Value></Value>
                    </Values>
                </Attribute>
            </Attributes>
        </SubRole>
    </SubRoles>
</ParentRole>
```

ParentRole

Element	Beschreibung	Mögliche Werte
`ParentId`	UUID der zusammengesetzten Rolle	UUID
`SubRoles`	Rollendefinition der untergeordneten Rollen	Das Element enthält folgende Elemente: • `SubRole` (Mehrfachdeklaration möglich)

SubRoles

Element	Beschreibung	Mögliche Werte
`SubRoles`	Rollendefinition der untergeordneten Rollen	Das Element enthält folgende Elemente: • `SubRole` (Mehrfachdeklaration möglich)

Element	Beschreibung	Mögliche Werte
`SubRole`	Rolle	Das Element enthält folgende Elemente: • `Id` • `Name` • `Description` (optional) • `Composite` (optional) • `ContainerId` • `Attributes` (optional)
`Id`	UUID der Rolle	Beliebiger String
`Name`	Name der Rolle	Beliebiger String
`Description`	Beschreibung der Rolle	Beliebiger String
`Composite`	Angabe, ob die Rolle eine zusammengesetzte Rolle ist	• `true`: Die Rolle ist eine zusammengesetzte Rolle. • `false`: Die Rolle ist keine zusammengesetzte Rolle. (Standard)
`ContainerId`	Name des Realm	Beliebiger String
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)

Attributes

Element	Beschreibung	
`Attributes`	Personalisierte, konfigurierbare Attribute	Das Element enthält folgende Elemente: • `Attribute` (Mehrfachdeklaration möglich)
`Attribute`	Personalisiertes, konfigurierbares Attribut	Das Element enthält folgende Elemente: • `Name` • `Values`
`Name`	Attributname	Beliebiger String
`Values`	Attributwerte	Das Element enthält folgende Elemente: • `Value` (Mehrfachdeklaration möglich)
`Value`	Attributwert	Beliebiger String

10.21.3.3.7.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="ParentRole">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="ParentId"/>
                <xs:element ref="SubRoles"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ParentId" type="xs:string"/>
    <xs:element name="SubRoles">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="SubRole"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="SubRole">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Id"/>
                <xs:element ref="Name"/>
                <xs:element ref="Attributes"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Id" type="xs:string"/>
    <xs:element name="Attributes">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Attribute"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Attribute">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Name"/>
                <xs:element maxOccurs="unbounded" ref="Values"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Values">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Value"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
```

```
    <xs:element name="Value" type="xs:string"/>
    <xs:element name="Name" type="xs:string"/>
</xs:schema>
```

10.21.3.3.7.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<ParentRole>
    <ParentId>3915229f-7544-4701-b1dc-6092861d9101</ParentId>
    <SubRoles>
        <SubRole>
            <Id>4915229f-7544-4701-b1dc-6092861d9102</Id>
            <Name>x4_admin_access_1</Name>
            <Attributes>
                <Attribute>
                    <Name>Team</Name>
                    <Values>
                        <Value>Red</Value>
                    </Values>
                    <Values>
                        <Value>Blue</Value>
                    </Values>
                </Attribute>
            </Attributes>
        </SubRole>
        <SubRole>
            <Id>5915229f-7544-4701-b1dc-6092861d9103</Id>
            <Name>x4_admin_access_2</Name>
            <Attributes>
                <Attribute>
                    <Name>Team</Name>
                    <Values>
                        <Value>Pink</Value>
                    </Values>
                    <Values>
                        <Value>Green</Value>
                    </Values>
                </Attribute>
            </Attributes>
        </SubRole>
    </SubRoles>
</ParentRole>
```

10.21.3.3.8 Get Composite of Role

Die Operation `Get Composite of Role` gibt die untergeordneten Rollen einer zusammengesetzten Rolle mit der angegebenen ID aus.

10.21.3.3.8.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Beschreibung	Mögliche Werte
Id	UUID der Rolle	UUID

10.21.3.3.8.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Id" type="xs:string"/>
</xs:schema>
```

10.21.3.3.8.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>3915229f-7544-4701-b1dc-6092861d9101</Id>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Roles>
    <Role>
        <Id>6342e31d-64f7-4fee-9441-c2be8f5db579</Id>
        <Name>x4_admin_access_1</Name>
        <Composite>false</Composite>
        <ClientRole>false</ClientRole>
        <ContainerId>X4Realm</ContainerId>
        <Attributes/>
    </Role>
    <Role>
        <Id>3915229f-7544-4701-b1dc-6092861d9101</Id>
        <Name>x4_dev_access_2</Name>
        <Composite>false</Composite>
        <ClientRole>false</ClientRole>
        <ContainerId>X4Realm</ContainerId>
        <Attributes/>
    </Role>
</Roles>
```

10.21.3.3.9 Add Roles to User

Die Operation `Add Roles to User` weist einem Benutzer eine oder mehrere Rollen zu.

10.21.3.3.9.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<User>
    <UserId></UserId>
    <Roles>
        <Role>
            <Name></Name>
            <Id></Id>
        </Role>
    </Roles>
</User>
```

User

Element	Beschreibung	Mögliche Werte
`UserId`	UUID des Benutzers	UUID
`Roles`	Rollendefinitionen	Das Element enthält folgende Elemente: • `Role` (Mehrfachdeklaration möglich)

Roles

Element	Beschreibung	Mögliche Werte
`Roles`	Rollendefinitionen	Das Element enthält folgende Elemente: • `Role` (Mehrfachdeklaration möglich)
`Role`	Rollendefinition	Das Element enthält folgende Elemente: • `Id` • `Name`
`Id`	UUID der Rolle	UUID
`Name`	Name der Rolle	Beliebiger String

10.21.3.3.9.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="User">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="UserId"/>
                <xs:element ref="Roles"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="UserId" type="xs:string"/>
    <xs:element name="Roles">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Role"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Role">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Id"/>
                <xs:element ref="Name"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Id" type="xs:string"/>
    <xs:element name="Name" type="xs:string"/>
</xs:schema>
```

10.21.3.3.9.3 Beispiele

Input

```
<User>
    <UserId>f37db159-8c3c-4b0a-ae7f-5362dcf4ed1b</UserId>
    <Roles>
        <Role>
            <Id>3915229f-7544-4701-b1dc-6092861d9101</Id>
            <Name>x4_dev_access</Name>
        </Role>
    </Roles>
</User>
```

10.21.3.3.10 Get Users with Role

Die Operation `Get Users with Role` gibt die Daten der Benutzer aus, die der angegebenen Rolle zugewiesen sind.

10.21.3.3.10.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Name></Name>
```

Name

Element	Beschreibung	Mögliche Werte
Name	Name der Rolle	Beliebiger String

10.21.3.3.10.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Name" type="xs:string"/>
</xs:schema>
```

10.21.3.3.10.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Name>x4_dev_access</Name>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Users>
    <User>
        <Id>1e6d7aed-8d82-4926-bf8d-fc490002f377</Id>
        <CreatedTimestamp>1634287963141</CreatedTimestamp>
        <Username>john</Username>
        <Enabled>true</Enabled>
        <Totp>false</Totp>
        <EmailVerified>true</EmailVerified>
        <FirstName>John</FirstName>
        <LastName>Doe</LastName>
        <Email>john@example.com</Email>
        <Attributes/>
        <NotBefore>0</NotBefore>
    </User>
    <User>
        <Id>a7bbfc23-a129-47c6-a802-e748d73876a6</Id>
        <CreatedTimestamp>1634287894774</CreatedTimestamp>
        <Username>jane</Username>
        <Enabled>true</Enabled>
        <Totp>false</Totp>
        <EmailVerified>true</EmailVerified>
        <FirstName>Jane</FirstName>
        <LastName>Doe</LastName>
        <Email>jane@example.com</Email>
        <Attributes/>
        <NotBefore>0</NotBefore>
    </User>
</Users>
```

10.21.3.3.11 Remove Roles from a User

Die Operation `Remove Roles from a User` entfernt zugewiesene Rollen des angegebenen Benutzers.

10.21.3.3.11.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<User>
    <UserId></UserId>
    <Roles>
        <Role>
            <Id></Id>
            <Name></Name>
        </Role>
    </Roles>
</User>
```

User

Element	Beschreibung	Mögliche Werte
`User`	Benutzerdefinition	Das Element enthält folgende Elemente: • `UserId` • `Roles`
`UserId`	UUID des Benutzers	UUID
`Roles`	Rollendefinitionen	Das Element enthält folgende Elemente: • `Role` (Mehrfachdeklaration möglich)

Roles

`Roles`	Rollendefinitionen	Das Element enthält folgende Elemente: • `Role` (Mehrfachdeklaration möglich)
`Role`	Rollendefinition	Das Element enthält folgende Elemente: • `Id` • `Name`
`Id`	UUID der Rolle	UUID
Element	**Beschreibung**	**Mögliche Werte**
`Name`	Name der Rolle	Beliebiger String

10.21.3.3.11.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="User">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="UserId"/>
                <xs:element ref="Roles"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="UserId" type="xs:string"/>
    <xs:element name="Roles">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Role"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Role">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Name"/>
                <xs:element ref="Id"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Id" type="xs:string"/>
</xs:schema>
```

10.21.3.3.11.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<User>
    <UserId>f37db159-8c3c-4b0a-ae7f-5362dcf4ed1b</UserId>
    <Roles>
        <Role>
            <Name>Developer</Name>
            <Id>2d725ff5-4da7-4f84-9224-d641499723de</Id>
        </Role>
        <Role>
            <Name>Software Architect</Name>
            <Id>78f84034-3129-4636-97c5-189b338deeb4</Id>
        </Role>
    </Roles>
</User>
```

10.21.3.3.12 Add Roles to Group

Die Operation `Add Roles to Group` weist einer Gruppe eine oder mehrere Rollen zu.

10.21.3.3.12.1 Input-XML

```
<Group>
    <GroupId></GroupId>
    <Roles>
        <Role>
            <Id></Id>
            <Name></Name>
        </Role>
    </Roles>
</Group>
```

Group

Element	Beschreibung	Mögliche Werte
`Group`	Gruppendefinition	Das Element enthält folgende Elemente: • `GroupId` • `Roles`
`GroupId`	UUID der Gruppe	UUID
`Roles`	Rollendefinitionen	Das Element enthält folgende Elemente: • `Role` (Mehrfachdeklaration möglich)

Roles

Element	Beschreibung	Mögliche Werte
`Roles`	Rollendefinitionen	Das Element enthält folgende Elemente: • `Role` (Mehrfachdeklaration möglich)
`Role`	Rollendefinition	Das Element enthält folgende Elemente: • `Id` • `Name`
`Id`	UUID der Rolle	UUID
`Name`	Name der Rolle	Beliebiger String

10.21.3.3.12.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Group">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="GroupId"/>
                <xs:element ref="Roles"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GroupId" type="xs:string"/>
    <xs:element name="Roles">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Role"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Role">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Id"/>
                <xs:element ref="Name"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Id" type="xs:string"/>
    <xs:element name="Name" type="xs:string"/>
</xs:schema>
```

10.21.3.3.12.3 Beispiele

Input

```
<Group>
    <GroupId>8445918e-f3ba-4c93-b3c4-7afd7b65fa57</GroupId>
    <Roles>
        <Role>
            <Id>3915229f-7544-4701-b1dc-6092861d9101</Id>
            <Name>x4_dev_access</Name>
        </Role>
    </Roles>
</Group>
```

10.21.3.3.13 Remove Roles From a Group

Die Operation `Remove Roles from a Group` entfernt zugewiesene Rollen aus der angegebenen Gruppe.

10.21.3.3.13.1 Input-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Group>
    <GroupId></GroupId>
    <Roles>
        <Role>
            <Id></Id>
            <Name></Name>
        </Role>
    </Roles>
</Group>
```

Group

Element	Beschreibung	Mögliche Werte
`Group`	Gruppendefinition	Das Element enthält folgende Elemente: • `GroupId` • `Roles`
`GroupId`	UUID der Gruppe	UUID
`Roles`	Rollendefinitionen	Das Element enthält folgende Elemente: • `Role` (Mehrfachdeklaration möglich)

Roles

`Roles`	Rollendefinitionen	Das Element enthält folgende Elemente: • `Role` (Mehrfachdeklaration möglich)
`Role`	Rollendefinition	Das Element enthält folgende Elemente: • `Id` • `Name`
`Id`	UUID der Rolle	UUID
Element	**Beschreibung**	**Mögliche Werte**
`Name`	Name der Rolle	Beliebiger String

10.21.3.3.13.2 XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="Group">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="GroupId"/>
                <xs:element ref="Roles"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="GroupId" type="xs:string"/>
    <xs:element name="Roles">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" ref="Role"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Role">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Name"/>
                <xs:element ref="Id"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Id" type="xs:string"/>
</xs:schema>
```

10.21.3.3.13.3 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Group>
    <GroupId>8445918e-f3ba-4c93-b3c4-7afd7b65fa57</GroupId>
    <Roles>
        <Role>
            <Name>Developer</Name>
            <Id>2d725ff5-4da7-4f84-9224-d641499723de</Id>
        </Role>
        <Role>
            <Name>Software Architect</Name>
            <Id>78f84034-3129-4636-97c5-189b338deeb4</Id>
        </Role>
    </Roles>
</Group>
```

10.21.4 Process Monitoring

`Process Monitoring`: Listet alle Prozessschritte und Ausführungszeiten eines Prozesses auf und gibt detaillierte Informationen zu einem bestimmten Prozess aus.

> **Hinweis:**
> Im Output des Adapters sind die Informationen enthalten, die zum Zeitpunkt des Outputs im Profiling vorhanden sind. Je nach Konfiguration werden die Profiling-Daten nach einem bestimmten Zeitraum gelöscht. Falls längere Zeiträume berücksichtigt werden sollen und Persistenz gewünscht ist, muss dies entsprechend konfiguriert werden. Weitere Informationen dazu finden Sie im X4 Control Center-Handbuch > Process Monitor > Profiling.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `List Process Executions`: Alle Prozesse und Ausführungszeiten eines Prozesses ausgeben • `Find Process Execution Detail`: Detaillierte Informationen zu einem Prozess mit der angegebenen UUID ausgeben • `List Process Execution Steps Detail`: Prozessschritte eines Prozesses mit der angegebenen UUID ausgeben

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.plugins.profiler.ProcessProfilerAdapter*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Weitere Informationen:

- Operation: Find Process Execution Detail
- Operation: List Process Executions
- Operation: List Process Execution Steps Detail

10.21.4.1 Operation: Find Process Execution Detail

Die Operation `Find Process Execution Detail` gibt detaillierte Informationen zu einem Prozess mit der angegebenen PID aus.

10.21.4.1.1 Beispiele

10.21.4.1.1.1 Input

Der Adapter erwartet ein XML-Dokument als Input:

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>15c28be0-d02b-4ca9-be08-b4096afef7f6</Id>
```

Folgende Elemente müssen im Input enthalten sein:

Element	Beschreibung
Id	UUID des Prozesses

10.21.4.1.1.2 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProcessInformation id="541109a1-b5ec-40f8-bc9f-a19e33a0564e">
    <HostName>DESKTOP-MRJDMVA</HostName>
    <Process>1/ESBProject_1/Processes/process.wrf</Process>
    <StartedAt>2022-12-02T12:06:37.279955Z</StartedAt>
    <FinishedAt>2022-12-02T12:06:41.17392Z</FinishedAt>
    <TotalTime>3893</TotalTime>
    <Status>FINISHED</Status>
    <DebugMode>true</DebugMode>
</ProcessInformation>
```

Weitere Informationen finden Sie im Abschnitt

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.4.2 Operation: List Process Executions

Die Operation `List Process Executions` listet Prozesse und Ausführungszeiten eines Prozesses auf.

Als Input muss ein Filter angegeben werden.

Der Adapter erwartet als Input-XML folgenden Aufbau:

```
<?xml version="1.0" encoding="UTF-8"?>
<Query>
  <Paging limit="" offset=""/>
  <!-- FILTER -->
</Query>
```

Folgende Elemente sind im Input enthalten:

Element	Beschreibung
`Paging`	Paging wird verwendet, um Daten nur in Teilmengen auszugeben.
`Query`	Filter

Weitere Informationen finden Sie im Abschnitt

- Nach allen Subprozessen eines Prozesses filtern
- Nach Prozesspfad filtern
- Nach Prozessstatus filtern
- Nach Subprozessen eines Prozesses filtern
- Nach Zeitraum filtern
- Daten im Output sortieren
- Daten in Teilmengen ausgeben

10.21.4.2.1 Nach allen Subprozessen eines Prozesses filtern

Um nach allen Subprozessen eines Prozesses einschließlich der Subprozesse der Subprozesse zu filtern, können folgende Elemente deklariert werden:

Element	Beschreibung
`ProcessFilterByRootId`	Filtertyp **Enthält folgende Elemente:** • `ProcessStatus` (optional) • `RootProcessInstanceId`
`ProcessStatus`	Prozessstatus, mehrfache Deklaration möglich. **Mögliche Werte:** • `RUNNING` • `PAUSED` • `FINISHED` • `ERROR` • `TERMINATED` • `CRASHED`
`RootProcessInstanceId`	UUID des Prozesses

10.21.4.2.1.1 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Query>
    <ProcessFilterByRootId>
        <ProcessStatus>FINISHED</ProcessStatus>
        <RootProcessInstanceId>5acfe639-f8be-44f7-8775-002e7dc6385a
        </RootProcessInstanceId>
    </ProcessFilterByRootId>
</Query>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProcessExecutions>
    <Paginate offset="0" limit="2"/>
    <ProcessExecution id="14336319-9e13-4dde-910f-282d23817551" rootId="5acfe639-
f8be-44f7-8775-002e7dc6385a">
        <HostName>SOFTPROJECT</HostName>
        <ParentProcess>
            <ProcessInstanceId>5acfe639-f8be-44f7-8775-002e7dc6385a</
ProcessInstanceId>
            <ActionId>3</ActionId>
        </ParentProcess>
        <Process>1/MyProject/Processes/MyProcess.wrf/MySubProcess.wrf</Process>
        <StartedAt>2021-09-06T11:30:34Z</StartedAt>
        <FinishedAt>2021-09-06T11:30:36Z</FinishedAt>
        <TotalTime>2005</TotalTime>
        <Status>FINISHED</Status>
    </ProcessExecution>
    <ProcessExecution id="a4a2298e-a952-41c3-81c1-d16d2066317a" rootId="5acfe639-
f8be-44f7-8775-002e7dc6385a">
        <HostName>SOFTPROJECT</HostName>
        <ParentProcess>
            <ProcessInstanceId>14336319-9e13-4dde-910f-282d23817551</
ProcessInstanceId>
            <ActionId>3</ActionId>
        </ParentProcess>
        <Process>1/MyProject/Processes/MyProcess.wrf/MySubProcess.wrf/
MySubProcess_2.wrf</Process>
        <StartedAt>2021-09-06T11:30:34Z</StartedAt>
        <FinishedAt>2021-09-06T11:30:35Z</FinishedAt>
        <TotalTime>1003</TotalTime>
        <Status>FINISHED</Status>
    </ProcessExecution>
</ProcessExecutions>
```

10.21.4.2.2 Nach Prozesspfad filtern

Um nach dem Prozesspfad zu filtern, müssen folgende Elemente deklariert werden:

Element	Beschreibung
`ProcessFilterByPath`	Filtertyp **Enthält folgende Elemente:** • `ProcessStatus` (optional) • `ProcessPath`

Element	Beschreibung
`ProcessStatus`	Prozessstatus, mehrfache Deklaration möglich. **Mögliche Werte:** • `RUNNING` • `PAUSED` • `FINISHED` • `ERROR` • `TERMINATED` • `CRASHED`
`ProcessPath`	Pfad zum auszuführenden X4-Prozess **Mögliche Werte:** Beliebig gültiger Pfad zum X4-Prozess im X4 Repository (z. B. `1/MyProject/Processes/MyProcess_1.wrf`)

10.21.4.2.2.1 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Query>
  <ProcessFilterByPath>
    <ProcessPath>1/MyProject/Processes/MyProcess.wrf</ProcessPath>
  </ProcessFilterByPath>
</Query>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProcessExecutions>
    <Paginate offset="0" limit="10"/>
    <ProcessExecution id="801cb04f-e501-4d73-8cff-9d3103ebf386">
        <HostName>SOFTPROJECT</HostName>
        <Process>1/MyProject/Processes/MyProcess.wrf</Process>
        <StartedAt>2021-09-02T15:50:28Z</StartedAt>
        <Status>RUNNING</Status>
    </ProcessExecution>
</ProcessExecutions>
```

10.21.4.2.3 Nach Prozessstatus filtern

Um nach dem Prozessstatus zu filtern, müssen folgende Elemente deklariert werden:

Element	Beschreibung
`ProcessFilter`	Filtertyp **Enthält folgende Elemente:** • `ProcessStatus`
`ProcessStatus`	Prozessstatus, mehrfache Deklaration möglich. **Mögliche Werte:** • RUNNING • PAUSED • FINISHED • ERROR • TERMINATED • CRASHED

10.21.4.2.3.1 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Query>
  <ProcessFilter>
    <ProcessStatus>RUNNING</ProcessStatus>
    <ProcessStatus>FINISHED</ProcessStatus>
  </ProcessFilter>
</Query>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProcessExecutions>
    <Paginate offset="0" limit="10"/>
    <ProcessExecution id="801cb04f-e501-4d73-8cff-9d3103ebf386">
        <HostName>SOFTPROJECT</HostName>
        <Process>1/MyProject/Processes/MyProcess.wrf</Process>
        <StartedAt>2021-09-02T15:50:28Z</StartedAt>
        <Status>RUNNING</Status>
    </ProcessExecution>
</ProcessExecutions>
```

10.21.4.2.4 Nach Subprozessen eines Prozesses filtern

Um nach Subprozessen eines Prozesses zu filtern, können folgende Elemente deklariert werden:

Element	Beschreibung
`ProcessFilterByParentProcess`	Filtertyp **Enthält folgende Elemente:** • `ProcessStatus` (optional) • `CallingProcessReference`
`ProcessStatus`	Prozessstatus, mehrfache Deklaration möglich. **Mögliche Werte:** • `RUNNING` • `PAUSED` • `FINISHED` • `ERROR` • `TERMINATED` • `CRASHED`
`CallingProcessReference`	Filterdefinition **Enthält folgende Elemente:** • `ProcessInstanceId` • `ActionId`
`ProcessInstanceId`	UUID des Prozesses
`ActionId`	ID des Prozessschrittes ⓘ Die `ActionId` kann aus dem Output der Operation `Find Process Execution Detail` entnommen werden.

10.21.4.2.4.1 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Query>
  <ProcessFilterByParentProcess>
    <ProcessStatus>FINISHED</ProcessStatus>
    <CallingProcessReference>
      <ProcessInstanceId>cc0f0227-6f91-414d-baac-090dbc883e31</ProcessInstanceId>
      <ActionId>0</ActionId>
    </CallingProcessReference>
  </ProcessFilterByParentProcess>
</Query>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProcessExecutions>
    <Paginate offset="0" limit="1"/>
    <ProcessExecution id="84af6605-601a-4a46-a708-ae32c3125c87" rootId="cc0f0227-6f91
-414d-baac-090dbc883e31">
        <HostName>SOFTPROJECT</HostName>
        <ParentProcess>
            <ProcessInstanceId>cc0f0227-6f91-414d-baac-090dbc883e31</
ProcessInstanceId>
            <ActionId>3</ActionId>
        </ParentProcess>
        <Process>1/MyProject/Processes/MyProcess.wrf/MySubProcess.wrf</Process>
        <StartedAt>2021-09-06T10:59:28Z</StartedAt>
        <FinishedAt>2021-09-06T10:59:29Z</FinishedAt>
        <TotalTime>1006</TotalTime>
        <Status>FINISHED</Status>
    </ProcessExecution>
</ProcessExecutions>
```

10.21.4.2.5 Nach Zeitraum filtern

Um nach einem Zeitraum zu filtern, können folgende Elemente deklariert werden:

Element	Beschreibung
`Period`	Angabe des Zeitraums (gemäß ISO 8601) **Erforderliche Attribute:** • `from`: Startdatum des Zeitraums • `to`: Enddatum des Zeitraums

10.21.4.2.5.1 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Query>
  <Period from="2021-09-02T00:00:01Z" to="2021-09-02T17:29:59Z" />
</Query>
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProcessExecutions>
    <Paginate offset="0" limit="10"/>
    <ProcessExecution id="801cb04f-e501-4d73-8cff-9d3103ebf386">
        <HostName>SOFTPROJECT</HostName>
        <Process>1/MyProject/Processes/MyProcess.wrf</Process>
        <StartedAt>2021-09-02T15:50:28Z</StartedAt>
        <Status>RUNNING</Status>
    </ProcessExecution>
</ProcessExecutions>
```

10.21.4.2.6 Daten im Output sortieren

Sie können die Daten im Output nach bestimmten Kriterien entweder aufsteigend oder absteigend sortieren.

Die Sortierung aktivieren Sie, indem Sie in der Input-XML das Element `<Sorting>` deklarieren. Innerhalb des Elements `<Sorting>` werden die Kriterien der Sortierung im Element `<Field>` festgelegt.

Element	Beschreibung
`Sorting`	Sortierung **Enthält folgende Elemente:** • `Field`
`Field`	**Enthält folgende Elemente:** • `Name` • `Order`
`Name`	Kriterium der Sortierung **Mögliche Werte:** • `StartedAt`: Startdatum • `FinishedAt`: Enddatum • `ProcessInstanceId`: ID des Prozesses • `ProcessPath`: Prozesspfad • `ParentProcessInstanceId`: ID des übergeordneten Prozesses • `ParentActionId`: ID des übergeordneten Prozessschrittes • `RootProcessId`: ID des Stammprozesses • `HostName`: Hostname • `TotalTime`: Gesamtdauer • `Status`: Status

Element	Beschreibung
`Order`	Reihenfolge der Sortierung **Mögliche Werte:** • `ASC`: Aufsteigend • `DESC`: Absteigend

10.21.4.2.6.1 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Query>
  <ProcessFilterByPath>
    <ProcessPath>1/MyProject/Processes/MyProcess.wrf</ProcessPath>
  </ProcessFilterByPath>
  <Sorting>
    <Field>
      <Name>StartedAt</Name>
      <Order>ASC</Order>
    </Field>
  </Sorting>
</Query>
```

10.21.4.2.7 Daten in Teilmengen ausgeben

Sie können die Daten in Teilmengen ausgeben, um z. B. die Rechenlast zu minimieren.

Deklarieren Sie dazu in der Input-XML das Element `<Paging>`. Die Teilmenge konfigurieren Sie mit den Attributen des Elements.

Attribut	Beschreibung
`offset`	Gibt den Offset der Datenabfrage an, also ab welchem Datensatz die Datenquelle ausgelesen werden soll. **Mögliche Werte:** Ganzzahl
`limit`	Anzahl der Datensätze, die in der Antwort enthalten sein sollen. **Mögliche Werte:** Ganzzahl größer als `0`

10.21.4.2.7.1 Beispiele

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Query>
  <Paging limit="10" offset="0"/>
  <ProcessFilterByRootId>
    <ProcessStatus>FINISHED</ProcessStatus>
    <RootProcessInstanceId>5acfe639-f8be-44f7-8775-002e7dc6385a
    </RootProcessInstanceId>
  </ProcessFilterByRootId>
</Query>
```

10.21.4.3 Operation: List Process Execution Steps Detail

Die Operation `List Process Execution Steps Detail` gibt Prozessschritte eines Prozesses mit der angegebenen UUID aus.

10.21.4.3.1 Beispiele

10.21.4.3.1.1 Input

Der Adapter erwartet als Input-XML folgenden Aufbau:

```
<?xml version="1.0" encoding="UTF-8"?>
<StepQuery>
  <ProcessInstanceId></ProcessInstanceId>
  <StepFilter>
    <TechnicalName></TechnicalName>
    <DisplayName></DisplayName>
    <Status></Status>
  </StepFilter>
  <Paging limit="" offset=""/>
</StepQuery>
```

Folgende Elemente sind im Input enthalten:

StepQuery

Element	Beschreibung
`ProcessInstanceId`	UUID des Prozesses
`StepFilter`	Filter (optional)
`Paging`	Paging wird verwendet, um Daten nur in Teilmengen auszugeben. (Optional)

StepFilter

Element	Beschreibung
`TechnicalName`	Technischer Name des Prozessbausteins

Element	Beschreibung
`DisplayName`	Label des Prozessbausteins
`Status`	Statuswert des Prozessbausteins

10.21.4.3.1.2 Output

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessExecutionStepsDetail id="e3956772-1e5b-48ce-8b16-e1c208713698">
   <Paginate limit="2" offset="0"/>
   <ProcessSteps>
      <ProcessStep>
         <ActionId>0</ActionId>
         <TechnicalName>Start</TechnicalName>
         <DisplayName>Start</DisplayName>
         <StartedAt>2022-11-04T10:12:25.689216Z</StartedAt>
         <FinishedAt>2022-11-04T10:12:25.690216Z</FinishedAt>
         <Status>0</Status>
         <Time>0</Time>
         <BytesOnPipe>0</BytesOnPipe>
         <DebugMode>true</DebugMode>
      </ProcessStep>
      <ProcessStep>
         <ActionId>4</ActionId>
         <TechnicalName>iXadMan</TechnicalName>
         <DisplayName>URL_Reader_1.fa</DisplayName>
         <StartedAt>2022-11-04T10:12:26.425029Z</StartedAt>
         <FinishedAt>2022-11-04T10:12:26.433028Z</FinishedAt>
         <Status>-1</Status>
         <ExceptionMessage>de.softproject.integration.adapter.urlReader.URLReader
from [Module "deployment.x4-7.3.0-SNAPSHOT.ear" from Service Module Loader]</
ExceptionMessage>
         <Time>7</Time>
         <BytesOnPipe>0</BytesOnPipe>
         <DebugMode>true</DebugMode>
      </ProcessStep>
   </ProcessSteps>
</ProcessExecutionStepsDetail>
```

10.21.5 Process Monitoring Management

`Process Monitoring Management`: API zum Ändern der Laufzeitkonfiguration für das Monitoring von X4 Prozessen.

Ein Monitoring definiert, für welche Prozesse innerhalb eines Projekts Informationen protokolliert werden. Somit können z. B. bestimmte Prozesse vom Logging ausgeschlossen werden.
In einem Projekt kann nur eine Monitoring-Datei (`.profiling`) erstellt werden.
Die Informationen zu ausgeführten und geloggten Prozessen können mit dem Adapter **Process Monitoring** ausgelesen werden.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `List Profilings`: Alle Profilings auflisten • `Get Profiling by Id`: Profiling mit der angegebenen UUID ausgeben • `Update Profiling`: Profiling mit der angegebenen UUID aktualisieren

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.core.profiling.runtime.ProfilingsAdapter*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Weitere Informationen:

- Operation List Profilings
- Operation Get Profiling by Id
- Operation Update Profiling

10.21.5.1 Operation List Profilings

Die Operation `List Profilings` listet alle Profilings innerhalb des X4 Repositorys auf.

10.21.5.1.1 Beispiele

10.21.5.1.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Profilings>
    <Profiling>
        <Id>53b4a859-8b3b-4007-a3b4-f2fa1c90090a</Id>
        <RepositoryName>1</RepositoryName>
        <ProjectName>MyProject</ProjectName>
    </Profiling>
    <!-- ... further Profiling entries ... -->
</Profilings>
```

Folgende Elemente können im Output enthalten sein:

Element	Beschreibung
Id	UUID des Projekts, in dem sich eine Profiling `.profiling` befindet.
RepositoryName	Name des X4 Repository-Ordners
ProjectName	Name des Projekts im X4 Repository

Weitere Informationen finden Sie im Abschnitt

- Operation List Profilings
- Operation Get Profiling by Id
- Operation Update Profiling

10.21.5.2 Operation Get Profiling by Id

Die Operation `Get Profiling by Id` gibt das Profiling aus, das sich in dem Projekt mit der angegeben Id befindet.

10.21.5.2.1 Beispiele

10.21.5.2.1.1 Input

Der Adapter erwartet ein XML-Dokument als Input:

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>f30c0ce7-fbc5-4988-81d4-7de74cb4dca8</Id>
```

Folgende Elemente müssen im Input enthalten sein:

Element	Beschreibung
Id	UUID des Projekts, in dem sich das Profling `.profiling` befindet.

10.21.5.2.1.2 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<Profiling active="true">
   <Id>f7adb9fe-93ea-4ca2-9023-1ebeca012930</Id>
   <Processes filterType="include">
      <Process path="Processes/dummy.wrf"/>
   </Processes>
</Profiling>
```

Folgende Elemente können im Output enthalten sein:

Element	Beschreibung
Id	UUID des Projekts, in dem sich das Profiling `.profiling` befindet.
Processes	Liste der Prozesse, für die Profiling-Regeln existieren.
@filterType	Attribut: Legt fest, ob die unterhalb aufgeführten Prozesse explizit in das Logging einbezogen oder ausgeschlossen werden. Mögliche Werte: • `include` • `exclude`
Process	Element mit relativem Pfad zum jeweiligen Prozess, für den Profiling-Regen gelten. Mehrfache Deklaration möglich.

Weitere Informationen finden Sie im Abschnitt

- Operation List Profilings
- Operation Get Profiling by Id
- Operation Update Profiling

10.21.5.3 Operation Update Profiling

Die Operation `Update Profiling` aktualisiert das Profiling, das sich in dem Projekt mit der angegeben Id befindet.

Die Änderungen werden nur zur Laufzeit aktualisiert. Die Datei vom Dateityp `.profiling` im Repository wird nicht aktualisiert.

10.21.5.3.1 Beispiele

10.21.5.3.1.1 Input

Der Adapter erwartet ein XML-Dokument als Input.

```
<?xml version="1.0" encoding="UTF-8"?>
<Profiling active="true">
    <Id>f7adb9fe-93ea-4ca2-9023-1ebeca012931</Id>
    <Processes filterType="include">
        <Process path="Processes/dummy.wrf" />
        <Process path="Processes/Subprocesses/dummy_subprocess.wrf" />
    </Processes>
</Profiling>
```

Folgende Elemente müssen im Input enthalten sein:

Element	Attribut	Beschreibung
`Profiling`	`active`	**Mögliche Werte:** • `true`: Aktiviert das Prozess-Profiling für das aktuelle Projekt. • `false`: Deaktiviert das Prozess-Profiling, identisch zum Zustand ohne `.profiling`-Datei innerhalb des jeweiligen Projekts.
`Id`		UUID des Projekts, in dem sich das Profiling `.profiling` befindet.
`Processes`		Eltern-Element zu einzeln darin aufgeführten `Process` Elementen. (optional) ⓘ Wird kein Processes angegeben, bezieht sich das Profiling mit dem filterType `include` auf alle Prozesse des aktuellen Projekts.
`Processes`	`filterType`	**Mögliche Werte:** • `include` Die folgende Prozessliste soll explizit geloggt werden, alle weiteren werden nicht mitgeloggt. • `exclude` Die folgende Prozessliste soll explizit vom Logging ausgeschlossen werden, alle weiteren werden geloggt. ⓘ Es ist nur die Angabe eines `Processes`-Elements mit entsprechend einem `filterType` möglich.
`Process`		Einzelner Prozess, für den Profiling-Regeln gelten. Mehrfache Deklaration möglich.
`Process`	`path`	Relativer Pfad innerhalb des jeweiligen Repository-Projekts. **Mögliche Werte:** Zeichenkette

✓ Weitere Informationen finden Sie im Abschnitt

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.6 Process Runtime Management

`Process Runtime Management`: Stoppt, beendet und listet laufende Prozesse auf.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `List Running Processes`: Alle laufenden Prozesse auflisten • `Stop Process`: Prozess mit angegebener UUID stoppen, der aktuelle Prozessschritt wird ausgeführt • `Terminate Process`: Prozess mit angegebener UUID versuchen unverzüglich zu beenden, der aktuelle Prozessschritt wird abgebrochen

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.plugins.management.ProcessRuntimeAdapter*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Weitere Informationen:

- Operation: List Running Processes
- Operation: Stop Process
- Operation: Terminate Process

10.21.6.1 Operation: List Running Processes

Die Operation `List Running Processes` listet alle auf dem X4 Server laufenden Prozesse auf.

10.21.6.1.1 Beispiele

10.21.6.1.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Processes>
    <Process id="d1539aa1-3493-4f0d-8a0f-cd0a2257f1f0">
        <Repository>1</Repository>
        <Project>Processes</Project>
        <Process>MyProcess.wrf</Process>
        <StartedAt>2021-08-24T06:00:00Z</StartedAt>
    </Process>
</Processes>
```

Folgende Elemente können im Output enthalten sein:

Element	Beschreibung
Process:id	UUID des laufenden Prozesses
Repository	Name des X4 Repository-Ordners
Project	Name des Ordners, in dem sich der Prozess im X4 Repository befindet
Process	Name des laufenden Prozesses
StartedAt	Startdatum und Startzeit des laufenden Prozesses

Weitere Informationen finden Sie im Abschnitt

- Operation: List Running Processes
- Operation: Stop Process
- Operation: Terminate Process

10.21.6.2 Operation: Stop Process

Die Operation `Stop Process` stoppt den laufenden Prozess mit der angegebenen PID. Der aktuelle Prozessschritt wird ausgeführt.

10.21.6.2.1 Beispiele

10.21.6.2.1.1 Input

Der Adapter erwartet ein XML-Dokument als Input.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<id>f744ffa1-960a-4122-a407-3dcdbd2a7924</id>
```

Folgende Elemente müssen im Input enthalten sein:

Element	Beschreibung
`Id`	UUID des laufenden Prozesses

Weitere Informationen finden Sie im Abschnitt

- Operation: List Running Processes
- Operation: Stop Process
- Operation: Terminate Process

10.21.6.3 Operation: Terminate Process

Die Operation `Terminate Process` versucht, den laufenden Prozess mit der angegebenen UUID unverzüglich zu beenden. Der aktuelle Prozessschritt wird abgebrochen.

10.21.6.3.1 Beispiele

10.21.6.3.1.1 Input

Der Adapter erwartet ein XML-Dokument als Input.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<id>f744ffa1-960a-4122-a407-3dcdbd2a7924</id>
```

Folgende Elemente müssen im Input enthalten sein:

Element	Beschreibung
`Id`	UUID des laufenden Prozesses

Weitere Informationen finden Sie im Abschnitt

- Operation: List Running Processes
- Operation: Stop Process
- Operation: Terminate Process

10.21.7 Project Parameter Management

`Project Parameter Management`: Dieser Adapter enthält ein API zum Auslesen und Ändern von Projekt-Parametern zur Laufzeit, so dass diese an die Systemumgebung angepasst werden können.

Eigenschaften (Properties)

`Operation`	Hier legen Sie fest, welche Operation der Adapter ausführt. **Mögliche Werte:** • `Get Project Parameter IDs`: Gibt die IDs von Projekten, die Parameter enthalten, mit Workspaces und Projekt-Namen aus • `Get Project Parameter Structure`: Gibt die XML-Struktur der Projekt-Parameter mit Gruppen aus • `List Project Parameters`: Gibt die XML-Struktur der Projekt-Parameter aus • `Patch Project Parameters`: Aktualisiert einen oder mehrere Projekt-Parameter für die angegebene Projekt-ID

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.core.parameter.runtime.ProjectParametrizationAdapter*: Hauptklasse (Standard)
`withSystemPrivileges`	**Mögliche Werte:** Boolean (true/false) False: Beschränkt den Zugriff zum Lesen und zum Bearbeiten von Parametern abhängig von den Zugriffsrechten (Rollen) True: Keine Einschränkung zum Lesen bzw. Bearbeiten von Parametern durch Zugriffsrechte (Rollen). Diese Einstellung muss genutzt werden, wenn ein Prozess nicht im Kontext eines Benutzers aufgerufen wird.

Statuswerte (prüfen)

`1 (successful)`	Die Operation wurde erfolgreich ausgeführt.
`0 (empty)`	Die Operation wurde erfolgreich ausgeführt, es wurde aber kein Parameter gefunden oder geändert.
`-1 (failed)`	Die Operation konnte nicht ausgeführt werden.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Weitere Informationen:

- Operation: Get Project Parameter Ids
- Operation: Get Project Parameter Structure
- Operation: List Project Parameters
- Operation: Patch Project Parameters

10.21.7.1 Operation: Get Project Parameter Ids

Die Operation `Get Project Parameter Ids` gibt die IDs der Projekte, die Parameter enthalten, mit Workspaces und Projekt-Namen aus.

Input

Der Adapter erwartet bei dieser Operation keinen Input.

10.21.7.1.1 Beispiel

Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProjectParameters xmlns:ns2="http://softproject.de/parameter/1.0">
   <ProjectParameter>
      <Id>f96a6444-2331-445d-ad56-83287df56e68</Id>
      <Workspace>1</Workspace>
      <ProjectName>Project_1</ProjectName>
   </ProjectParameter>
   <ProjectParameter>
      <Id>f86fa741-a4ad-40a8-88de-d0a279b592a3</Id>
      <Workspace>1</Workspace>
      <ProjectName>Project_2</ProjectName>
   </ProjectParameter>
   <ProjectParameter>
      <Id>316874e3-636b-4475-a6c4-e3601d998be9</Id>
      <Workspace>2</Workspace>
      <ProjectName>Project_1</ProjectName>
   </ProjectParameter>
   <ProjectParameter>
      <Id>410e950e-f48b-41a1-8d49-777fa15a9f67</Id>
      <Workspace>2</Workspace>
      <ProjectName>Project_2</ProjectName>
   </ProjectParameter>
</ProjectParameters>
```

Folgende Elemente können im Output enthalten sein:

Element	Attribut	Beschreibung
`ProjectParameter`		Parameter eines Projekts mit ID, Name und Workspace des Projekts
	`Id`	UUID des Projekts, in dem sich eine Datei `.param` mit Projekt-Parametern befindet.

Element	Attribut	Beschreibung
	`Workspace`	Name des Workspaces, in dem das Projekt gespeichert ist.
	`ProjectName`	Name des Projekts im X4 Repository

Die durch die Operation angezeigten Daten können in einem Mapping weiter verwertet werden (z.B. Auslesen der ID des aktuellen ESB-Projekts).

Weitere Informationen finden Sie in folgenden Abschnitten:

- Operation: Get Project Parameter Ids
- Operation: Get Project Parameter Structure
- Operation: List Project Parameters
- Operation: Patch Project Parameters

10.21.7.2 Operation: Get Project Parameter Structure

Die Operation `Get Project Parameters` erwartet eine `UUID` als Input und gibt die XML-Struktur der Projekt-Parameter mit Gruppen aus.

10.21.7.2.1 Beispiel

Input

Der Adapter erwartet folgende XML-Struktur als Input:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Id>efb66be5-9ff3-4c1a-9879-9f46854694c0</Id>
```

Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Parameters xmlns:ns2="http://softproject.de/parameter/1.0"
    id="efb66be5-9ff3-4c1a-9879-9f46854694c0" projectName="WebAppProject_2"
    workspace="1">
    <Enums/>
    <Group name="MyGroup">
        <Group name="MySubGroup">
            <VariableParameter name="subgroup" type="String">
                <Description>variable in subgroup</Description>
                <Value>b</Value>
            </VariableParameter>
        </Group>
        <VariableParameter name="test" type="String">
            <Description>variable in group</Description>
            <Value>a</Value>
        </VariableParameter>
    </Group>
    <VariableParameter name="test" type="String">
        <Description>variable on root level</Description>
        <Value>test</Value>
    </VariableParameter>
</Parameters>
```

Folgende Elemente können im Output enthalten sein:

Element	Attribut	Beschreibung
`Parameters`		Parameter eines Projects mit ID, Name und Workspace des Projekts
	`Id`	UUID des Projekts, in dem sich eine **Project Parameter Configuration** `.param` befindet.
	`ProjectName`	Name des Projekts im X4 Repository
	`Workspace`	Name des Workspaces, in dem das Projekt gespeichert ist.
`Enums`		
`Group name`		Name der Projekt-Parameter-Gruppe
`Variable parameter`		Projekt-Parameter (Variable)
`Constant parameter`		Projekt-Parameters (Constant)
	`completeName`	Name des Parameters mit Parameter-Gruppe
	`name`	Name des Parameters
	`type`	Datentyp des Projekt-Parameters
`Description`		Beschreibung des Projekt-Parameters
`Value`		Default-Wert des Projekt-Parameters

Weitere Informationen finden Sie in den folgenden Abschnitten:

- Operation: Get Project Parameter Ids
- Operation: Get Project Parameter Structure
- Operation: List Project Parameters
- Operation: Patch Project Parameters

10.21.7.3 Operation: List Project Parameters

Die Operation `List Project Parameters` erwartet eine `UUID` als Input und gibt die XML-Struktur der Projekt-Parameter für das Projekt mit dieser `UUID` aus.

10.21.7.3.1 Beispiel

Input

Der Adapter erwartet folgende XML-Struktur als Input:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Id>efb66be5-9ff3-4c1a-9879-9f46854694c0</Id>
```

Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Parameters xmlns:ns2="http://softproject.de/parameter/1.0"
            id="efb66be5-9ff3-4c1a-9879-9f46854694c0"
            projectName="WebAppProject_2"
            workspace="1">
<Variable completeName="MyGroup.Test">test</Variable>
</Parameters>
```

Folgende Elemente können im Output enthalten sein:

Element	Attribut	Beschreibung
`Parameters`		Parameter eines Projekts mit ID, Name und Workspace des Projekts
	`Id`	UUID des Projekts, in dem sich eine **Project Parameter Configuration** `.param` befindet.
	`ProjectName`	Name des Projekts im X4 Repository
	`Workspace`	Name des Workspaces, in dem das Projekt gespeichert ist.
`Variable`		Typ des Projekt-Parameters

Element	Attribut	Beschreibung
`Constant`		Typ des Projekt-Parameters
	`completeName`	Name des Parameters mit Parameter-Gruppe

Weitere Informationen finden Sie im Abschnitt

- Operation: Get Project Parameter Ids
- Operation: Get Project Parameter Structure
- Operation: List Project Parameters
- Operation: Patch Project Parameters

10.21.7.4 Operation: Patch Project Parameters

Die Operation `Patch Project Parameters` aktualisiert einen oder mehrere Projekt-Parameter für die angegebene Projekt-ID.

Die Änderungen werden nur zur Laufzeit aktualisiert. Die Datei vom Dateityp `.param` im Repository wird nicht aktualisiert.

Wenn die Operation **Patch Project Parameters** fehlschlägt, werden keine Parameter aktualisiert.

Aktualisierung von Variables und Constants
Projekt-Parameter vom Typ **Constant** können nicht aktualisiert werden.
Wenn Projekt-Parameter vom Typ **Variable** und **Constant** in der Input-XML-Datei enthalten sind, werden keine Parameter aktualisiert.

10.21.7.4.1 Beispiel

Input

Der Adapter erwartet ein XML-Dokument als Input.

```
<?xml version="1.0" encoding="UTF-8"?>
<Parameters xmlns:ns2="http://softproject.de/parameter/1.0"
            id="f96a6444-2331-445d-ad56-83287df56e68"
            projectName="Project_1"
            workspace="1">
   <Variable completeName="assignedExpert">assignedExpert: new value</Variable>
</Parameters>
```

Folgende Elemente müssen im Input enthalten sein:

Element	Attribut	Beschreibung
`Parameters`		Parameter eines Projects mit ID, Name und Workspace des Projekts
	`Id`	UUID des Projekts, in dem sich eine Project Parameter Configuration `.param` befindet.
	`ProjectName`	Name des Projekts im X4 Repository
	`Workspace`	Name des Workspaces, in dem das Projekt gespeichert ist.
`Variable`		Typ des Projekt-Parameters
`Constant`		Typ des Projekt-Parameters
	`completeName`	Name des Parameters mit Parameter-Gruppe

Output

Die Operation generiert einen Output mit den geänderten Parametern.

Weitere Informationen finden Sie im Abschnitt

- Operation: Get Project Parameter Ids
- Operation: Get Project Parameter Structure
- Operation: List Project Parameters
- Operation: Patch Project Parameters

10.21.8 Savepoint Management

`Savepoint Management`: Dieser Adapter enthält ein API für den Zugriff auf und die Arbeit mit Savepoints.

Eigenschaften (Properties)

`Operation`	Hier legen Sie fest, welche Operation der Adapter ausführt. **Mögliche Werte:** • `List Savepoints`: alle Savepoints auflisten • `Resume Savepoint`: Prozess nach dem Anhalten per Savepoint fortsetzen • `Delete Savepoint`: Savepoint löschen

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.plugins.management.SavePointAdapter*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation konnte nicht ausgeführt werden.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Weitere Informationen:

- Operation: List Savepoints
- Operation: Delete Savepoint
- Operation: Resume Savepoint

10.21.8.1 Operation: List Savepoints

Die Operation `List Savepoints` generiert eine Liste der in einem Prozess gesetzten Savepoints.

Input-Struktur

Der Adapter erwartet keinen Input.

10.21.8.1.1 Beispiel

Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Savepoints>
    <Savepoint id="2be2eab5-317e-4c02-9896-7de77c4fa217">
        <PID>16754187190436776 47</PID>
        <PUID>eab1040b-6635-4d64-9cbf-f613ace4291d</PUID>
        <Repository>1</Repository>
        <Project>SavePoints</Project>
        <Process>Processes/SaveMe.wrf</Process>
        <NextAction_ID>2</NextAction_ID>
    </Savepoint>
    <Savepoint id="72541d1c-fb40-433a-9cd3-677d8cdf814f">
        <PID>16754187190436776 47</PID>
        <PUID>eab1040b-6635-4d64-9cbf-f613ace4291d</PUID>
        <Repository>1</Repository>
        <Project>SavePoints</Project>
        <Process>Processes/SaveMe.wrf/SaveMePrivateSubprocess.wrf</Process>
        <NextAction_ID>1</NextAction_ID>
    </Savepoint>
    <Savepoint id="e2c06c03-1d66-4bdf-9f15-616ee9e8ce8b">
        <PID>16754187190436776 47</PID>
        <PUID>eab1040b-6635-4d64-9cbf-f613ace4291d</PUID>
        <Repository>1</Repository>
        <Project>SavePoints</Project>
        <Process>Processes/SaveMeSubprocess.wrf</Process>
        <NextAction_ID>3</NextAction_ID>
    </Savepoint>
</Savepoints>
```

Folgende Elemente können im Output enthalten sein:

Element	Beschreibung
`Savepoint`	Über das Attribut `id` wird die UUID des Savepoints angegeben.

<table>
<tr><th>Element</th><th>Beschreibung</th></tr>
<tr><td>PID</td><td>Prozess-ID

Hinweis:

Beim Abrufen der PID gibt es zwei Szenarien:

Szenario 1
Savepoints wird immer die PID des Root-Prozesses zugewiesen, wenn der Prozess, in dem ein Savepoint enthalten ist, über den Root-Prozess aufgerufen wird. Dieses Szenario ist im oben gezeigten Output dargestellt. Hier sind die PIDs für alle abgerufenen Savepoints identisch.

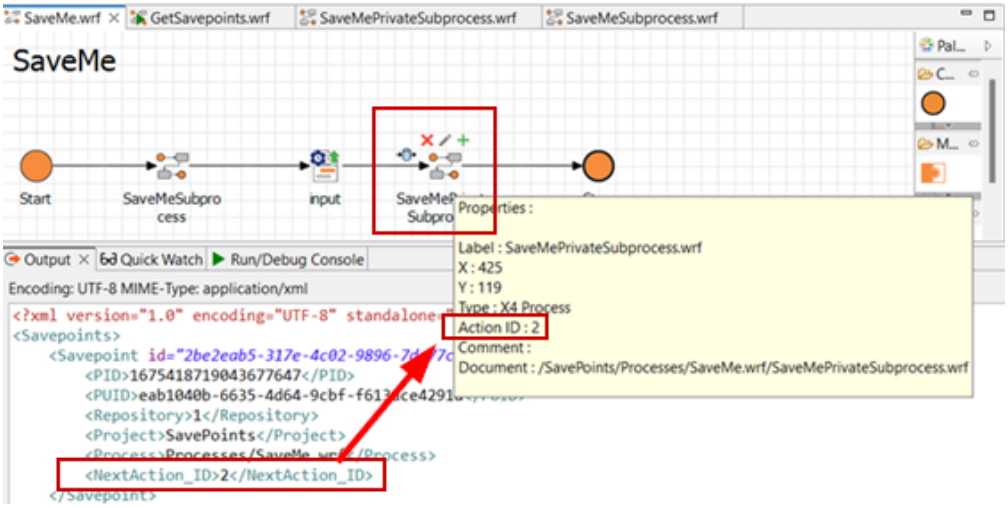
Szenario 2
Wird der Prozess mit dem Savepoint nicht mit dem Root-Prozess aufgerufen, wird die PID des Prozesses mit dem Savepoint verwendet.

Beispiel für Szenario 2:

<pre><?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Savepoints>
<Savepoint id="2be2eab5-317e-4c02-9896-7de77c4fa217">
<PID>16754187190436776 47</PID>
<PUID>eab1040b-6635-4d64-9cbf-f613ace4291d</PUID>
<Repository>1</Repository>
<Project>SavePoints</Project>
<Process>Processes/SaveMe.wrf</Process>
<NextAction_ID>2</NextAction_ID>
<Savepoint id="44d7e9b5-5fa2-403e-8645-dee669dc69d0">
<PID>16754187190549674 21</PID>
<PUID>88ffaa69-92e7-43d4-9ffc-3d36a2a50c2b</PUID>
<Repository>1</Repository>
<Project>SavePoints</Project>
<Process>Processes/SaveMeSubprocess.wrf</Process>
<NextAction_ID>3</NextAction_ID>
</Savepoint>
</Savepoints></pre></td></tr>
<tr><td>PUID</td><td>UUID des Prozesses, mit dem die Savepoints ausgelesen wurden

Hinweis:

Für Savepoints, die mit einer X4-Version vor 7.3 erstellt wurden, wird nur die PID ausgegeben, nicht die PUID.</td></tr>
<tr><td>Repository</td><td>Der Workspace, in dem das Projekt abgelegt ist</td></tr>
<tr><td>Project</td><td>Name des Projekts, in dem der Technical Process enthalten ist</td></tr>
<tr><td>Process</td><td>Name des Technical Processes, in dem die Savepoints gesetzt sind</td></tr>
</table>

Element	Beschreibung
`NextAction_ID`	Die Action ID des Elements, das auf den Savepoint folgt, der in dem im Output genannten Prozess gesetzt wurde Beispiel für den ersten Savepoint aus dem oben gezeigten Output: 

Weitere Informationen:

- Operation: List Savepoints
- Operation: Delete Savepoint
- Operation: Resume Savepoint

10.21.8.2 Operation: Delete Savepoint

Die Operation `Delete Savepoint` löscht einen gesetzten Savepoint.

Input-XML-Struktur

Der Adapter erwartet ein XML-Dokument als Input.

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Folgendes Element muss im Input enthalten sein:

Element	Beschreibung	Mögliche Werte
`Id`	UUID des Savepoints	UUID

10.21.8.2.1 Beispiel

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>cced145e-9056-4eb2-bbcf-756a92215c78</Id>
```

Output

Die Operation generiert keinen speziellen Output, d. h., der Output entspricht dem Input.

Weitere Informationen:

- Operation: List Savepoints
- Operation: Delete Savepoint
- Operation: Resume Savepoint

10.21.8.3 Operation: Resume Savepoint

Die Operation `Resume Savepoint` setzt einen Prozess nach dem Anhalten per Savepoint fort.

Input

Der Adapter erwartet ein XML-Dokument als Input.

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Folgendes Element muss im Input enthalten sein:

Element	Beschreibung	Mögliche Werte
`Id`	UUID des Savepoints	UUID

10.21.8.3.1 Beispiel

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>65ece1ae-182c-4c5b-9116-dc1f4c03355e</Id>
```

Output

Die Operation generiert keinen speziellen Output, d. h., der Output entspricht dem Input.

Weitere Informationen:

- Operation: List Savepoints
- Operation: Delete Savepoint
- Operation: Resume Savepoint

10.21.9 Schedule Management

`Schedule Management`: Aktualisiert oder listet Scheduler auf, die sich im X4 Repository oder in einem Projekt befinden.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `List Schedules`: Scheduler mit allen Zeitplänen auflisten • `Get Schedule by Id`: Scheduler mit der angegebenen Id ausgeben • `Update Schedule`: Scheduler mit der angegebenen Id aktualisieren • `Jobs Information`: Zeigt Informationen über zeitgesteuerte Prozesse aus allen Schedule-Dateien im X4 Repository an.

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.plugins.scheduler.runtime.SchedulesAdapter*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Weitere Informationen:

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.9.1 Operation: List Schedules

Die Operation `List Schedules` listet alle Schedules innerhalb des X4 Repositorys auf.

10.21.9.1.1 Beispiele

10.21.9.1.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Schedules>
    <Schedule>
        <Id>f30c0ce7-fbc5-4988-81d4-7de74cb4dca8</Id>
        <RepositoryName>1</RepositoryName>
        <ProjectName>MyProject</ProjectName>
    </Schedule>
</Schedules>
```

Folgende Elemente können im Output enthalten sein:

Element	Beschreibung
Id	UUID des Projekts, in dem sich der Scheduler `.schedule` befindet.
RepositoryName	Name des X4 Repository-Ordners
ProjectName	Name des Projekts im X4 Repository

Weitere Informationen finden Sie im Abschnitt

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.9.2 Operation: Get Schedule by Id

Die Operation `Get Schedule by Id` gibt den Scheduler aus, der sich in dem Projekt mit der angegeben Id befindet.

10.21.9.2.1 Beispiele

10.21.9.2.1.1 Input

Der Adapter erwartet ein XML-Dokument als Input:

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>f30c0ce7-fbc5-4988-81d4-7de74cb4dca8</Id>
```

Folgende Elemente müssen im Input enthalten sein:

Element	Beschreibung
Id	UUID des Projekts, in dem sich der Scheduler `.schedule` befindet.

10.21.9.2.1.2 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<Schedule>
   <Id>f30c0ce7-fbc5-4988-81d4-7de74cb4dca8</Id>
   <Cron>
      <Id>6dd6732e-15c1-4a63-bdfd-6d90d45eedb5</Id>
      <Name>Execute every minute</Name>
      <Process>Processes/ScheduledProcess.wrf</Process>
      <Active>true</Active>
      <Expression>0 6 6 ? * FRI *</Expression>
   </Cron>
   <Startup>
      <Id>a0b9dd4d-c722-4aa6-bc07-1f837d52c5d3</Id>
      <Name>Execution</Name>
      <Process>Processes/ScheduledProcess.wrf</Process>
      <Active>true</Active>
      <StartDate>2021-06-29T11:16:09.904+02:00</StartDate>
      <EndDate>2021-07-30T11:16:09.907+02:00</EndDate>
   </Startup>
</Schedule>
```

Folgende Elemente können im Output enthalten sein:

Element	Beschreibung
Id	UUID des Projekts, in dem sich der Scheduler `.schedule` befindet.
Cron	Ausführungstyp `Cron`
Startup	Ausführungstyp `Startup`
Startup:Id	UUID der Ausführung
Cron:Id	UUID der Ausführung

Element	Beschreibung
Name	Name der Ausführung
Process	Pfad zum Prozess, der ausgeführt werden soll
Active	Status der Ausführung **Mögliche Werte:** *true* / *false*
CronExpression	Cron-Ausdruck der Ausführung, nur für den Ausführungstyp `Cron` relevant
StartDate	Startdatum der Ausführung, nur für den Ausführungstyp `Startup` relevant
EndDate	Enddatum der Ausführung, nur für den Ausführungstyp `Startup` relevant

Weitere Informationen finden Sie im Abschnitt

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.9.3 Operation: Update Schedule

Die Operation `Update Schedule` aktualisiert den Scheduler, der sich in dem Projekt mit der angegeben Id befindet.

Die Änderungen werden nur zur Laufzeit aktualisiert. Die Datei vom Dateityp `.schedule` im Repository wird nicht aktualisiert.

10.21.9.3.1 Beispiele

10.21.9.3.1.1 Input

Der Adapter erwartet ein XML-Dokument als Input.

```
<?xml version="1.0" encoding="UTF-8"?>
<Schedule>
  <Id>f30c0ce7-fbc5-4988-81d4-7de74cb4dca8</Id>
    <Cron>
      <Id>6dd6732e-15c1-4a63-bdfd-6d90d45eedb5</Id>
      <Expression>0 6 6 ? * FRI *</Expression>
    </Cron>
</Schedule>
```

Folgende Elemente müssen im Input enthalten sein:

Element	Beschreibung
Id	UUID des Projekts, in dem sich der Scheduler `.schedule` befindet.

Element	Beschreibung
Cron	Ausführungstyp `Cron`
Startup	Ausführungstyp `Startup`
Startup:Id	UUID der Ausführung
Cron:Id	UUID der Ausführung
Active	Status der Ausführung **Mögliche Werte:** *`true`* / *`false`*
CronExpression	Cron-Ausdruck der Ausführung, nur für den Ausführungstyp `Cron` relevant
StartDate	Startdatum der Ausführung, nur für den Ausführungstyp `Startup` relevant
EndDate	Enddatum der Ausführung, nur für den Ausführungstyp `Startup` relevant

Weitere Informationen finden Sie im Abschnitt

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.9.4 Operation: Jobs Information

Die Operation `Jobs Information` zeigt Informationen über zeitgesteuerte Prozesse aus allen Schedule-Dateien im X4 Repository an.

10.21.9.4.1 Beispiele

10.21.9.4.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output.

Hinweis
Das Format für Zeit und Datum kann, abhängig von Betriebssystem und weiteren Einstellungen, unterschiedlich sein.
Mögliche Formate sind u.a.:

- DD.MM.YY, hh:mm
- MM/DD/YY, hh:mm AM/PM

Output im Format DD.MM.YY, hh:mm:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<JobInformation>
    <Job>
        <Id>5d6d4a9d-1815-426f-bf2c-fdb846d211e6</Id>
        <Name>ScheduledProcess</Name>
        <Process>Processes/MyProcess.wrf</Process>
        <Active>true</Active>
        <JobsExecuted>4</JobsExecuted>
        <StartTime>25.04.22, 10:30</StartTime>
        <PreviousTime>26.04.22, 10:30</PreviousTime>
        <NextTime>27.04.22, 10:30</NextTime>
    </Job>
</JobInformation>
```

Output im Format MM/DD/YY, hh:mm AM/PM:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<JobInformation>
    <Job>
        <Id>5d6d4a9d-1815-426f-bf2c-fdb846d211e6</Id>
        <Name>ScheduledProcess</Name>
        <Process>Processes/MyProcess.wrf</Process>
        <Active>true</Active>
        <JobsExecuted>4</JobsExecuted>
        <StartTime>4/25/22, 10:30 AM</StartTime>
        <PreviousTime>4/26/22, 10:30 AM</PreviousTime>
        <NextTime>4/27/22, 10:30 AM</NextTime>
    </Job>
</JobInformation>
```

Folgende Elemente können im Output enthalten sein:

Element	Beschreibung	Mögliche Werte
`Id`	UUID der Ausführung	• UUID
`Name`	Name der Ausführung	• Beliebiger String
`Process`	Pfad zum Prozess, der ausgeführt werden soll	• Beliebig gültiger Pfad zum X4-Prozess im X4 Repository
`Active`	Status der Ausführung	• `true` • `false`
`JobsExecuted`	Anzahl der bisherigen Ausführungen	• Ganzzahl
`StartTime`	Startdatum der Ausführung	• Zeit- und Datumsangabe
`PreviousTime`	Datum der letzten Ausführung	• Zeit- und Datumsangabe
`NextTime`	Datum der nächsten Ausführung	• Zeit- und Datumsangabe

Element	Beschreibung	Mögliche Werte
`EndTime`	Enddatum der Ausführung	• Zeit- und Datumsangabe

Weitere Informationen finden Sie im Abschnitt

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.10 System Configuration Management

`System Configuration Management`: Zeigt die Konfigurationsparameter der X4 Instanzen an.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `List Configurations`: Alle Konfigurationen des Systems auflisten • `Get Configuration by configurationName`: Konfiguration mit dem angegebenen Namen ausgeben

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.plugins.management.configuration.SystemConfigurationAdapter*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Weitere Informationen:

- Operation: List Configurations
- Operation: Get Configuration by configurationName

10.21.10.1 Operation: List Configurations

Die Operation `List Configurations` listet alle Konfigurationen des Systems auf.

10.21.10.1.1 Beispiele

10.21.10.1.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<SystemConfigurationFiles>
    <SystemConfigurationFile>
        <ConfigurationName>de.softproject.x4.auth.api.config.KeyCloakConfiguration</
ConfigurationName>
        <FileName>keycloak_config.json</FileName>
        <MediaType>application/json</MediaType>
        <Description>The configuration file of the authentication provider
Keycloak.</Description>
    </SystemConfigurationFile>
    <SystemConfigurationFile>
        <ConfigurationName>de.softproject.x4.plugins.sqlprofiling.database.config.Sql
ProfilingConfiguration</ConfigurationName>
        <FileName>sql_profiling_config.xml</FileName>
        <MediaType>application/xml</MediaType>
        <Description>The configuration file of the SQL Profiling.</Description>
    </SystemConfigurationFile>
</SystemConfigurationFiles>
```

10.21.10.2 Operation: Get Configuration by configurationName

Die Operation `Get Configuration by configurationName` gibt die Konfiguration mit dem angegebenen Namen aus.

10.21.10.2.1 Beispiele

10.21.10.2.1.1 Input

Der Adapter erwartet ein XML-Dokument als Input:

```
<?xml version="1.0" encoding="UTF-8"?>
<ConfigurationName>
    <Value>KeyCloakConfiguration</Value>
</ConfigurationName>
```

10.21.10.2.1.2 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Configuration>
    <FileDescription name="KeyCloakConfiguration">
        <Attributes>
            <Field name="connection">
                <Attributes>
                    <String name="realm" value="X4Realm"/>
                    <String name="resource" value="X4"/>
                    <String name="realmKey"/>
                    <String name="authServerUrl" value="http://localhost:8085/auth/"/
>
                    <String name="sslRequired"/>
                    <Integer name="confidentialPort" value="0"/>
                    <Boolean name="useResourceRoleMappings" value="false"/>
                    <Boolean name="cors" value="false"/>
                    <Integer name="corsMaxAge" value="-1"/>
                    <String name="corsAllowedMethods"/>
                    <String name="corsAllowedHeaders"/>
                    <String name="corsExposedHeaders"/>
                    <Boolean name="bearerOnly" value="false"/>
                    <Boolean name="autodetectBearerOnly" value="false"/>
                    <Boolean name="enableBasicAuth" value="false"/>
                    <Boolean name="exposeToken" value="false"/>
                    <Collection name="credentials">
                        <Entries/>
                    </Collection>
                    <Integer name="connectionPoolSize" value="20"/>
                    <Long name="socketTimeout" value="-1"/>
                    <Long name="connectionTimeout" value="-1"/>
                    <Long name="connectionTTL" value="-1"/>
                    <Boolean name="disableTrustManager" value="false"/>
                    <Boolean name="allowAnyHostname" value="false"/>
                    <String name="proxyUrl"/>
                    <String name="truststore"/>
                    <String name="truststorePassword"/>
                    <String name="clientKeystore"/>
                    <String name="clientKeystorePassword"/>
                    <String name="clientKeyPassword"/>
                    <Boolean name="alwaysRefreshToken" value="false"/>
                    <Boolean name="registerNodeAtStartup" value="false"/>
                    <Integer name="registerNodePeriod" value="-1"/>
                    <String name="tokenStore"/>
                    <String name="tokenCookiePath"/>
                    <String name="principalAttribute"/>
                    <Boolean name="turnOffChangeSessionIdOnLogin"/>
                    <Integer name="tokenMinimumTimeToLive" value="0"/>
                    <Integer name="minTimeBetweenJwksRequests" value="10"/>
                    <Integer name="publicKeyCacheTtl" value="86400"/>
                    <Boolean name="ignoreOAuthQueryParameter" value="false"/>
                    <Collection name="redirectRewriteRules">
                        <Entries/>
                    </Collection>
                    <Boolean name="verifyTokenAudience" value="false"/>
                    <Boolean name="publicClient" value="false"/>
```

```
                    <Boolean name="pkce" value="false"/>
                </Attributes>
            </Field>
            <Field name="restApiCredentials">
                <Attributes>
                    <String name="username" value="system"/>
                    <String name="password" value="******"/>
                </Attributes>
            </Field>
        </Attributes>
    </FileDescription>
</Configuration>
```

10.21.11 Third Party License Summary

`Third Party License Summary`: Listet die Lizenzinformationen der Bibliotheken von Drittanbietern auf.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Get Summary`: Lizenzinformationen der Bibliotheken von Drittanbietern auflisten

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.plugins.management.license.ThirdPartyLicenseAdapter*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Weitere Informationen:

- Operation: Get Summary

10.21.11.1 Operation: Get Summary

Die Operation `Get Summary` listet die Lizenzinformationen der Bibliotheken von Drittanbietern auf.

10.21.11.1.1 Beispiele

10.21.11.1.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ThirdParty>
    <Dependencies>
        <Dependency>
            <GroupId>org.infinispan</GroupId>
            <ArtefactId>infinispan-query-dsl</ArtefactId>
            <Version>12.1.7.Final</Version>
            <Licenses>
                <License>
                    <Name>Apache License 2.0</Name>
                    <URL>http://www.apache.org/licenses/LICENSE-2.0</URL>
                </License>
            </Licenses>
        </Dependency>
        ...
    </Dependencies>
</ThirdParty>
```

10.21.12 Web App Configuration Management

`Web App Configuration Management`: Aktualisiert oder listet Configuration-Dateien auf, die sich im X4 Repository oder in einem Projekt befinden.

Weitere Informationen zur Configuration in Web Apps finden Sie im **Web App-Handbuch**.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `List WebApp Configurations`: Liste mit allen Configurations im Repository ausgeben • `Get Configuration by Id`: Configuration mit der angegebenen Id ausgeben • `Update Configuration`: Configuration mit der angegebenen Id aktualisieren

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.server.webapp.adapter.WebAppConfigurationAdapter*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Weitere Informationen:

- Operation: Get Configuration by Id
- Operation: List WebApp Configurations
- Operation: Update Configuration

10.21.12.1 Operation: Get Configuration by Id

Die Operation `Get Configuration by Id` gibt die Configuration aus, die sich in dem Projekt mit der angegebenen Id befindet.

Wenn Änderungen zur Laufzeit vorgenommen wurden, wird der zur Laufzeit verwendete Inhalt der Configuration ausgegeben.

10.21.12.1.1 Beispiele

10.21.12.1.1.1 Input

Der Adapter erwartet ein XML-Dokument als Input:

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>0cbff222-12e8-43f3-a2b8-31388c50afee</Id>
```

Folgende Elemente müssen im Input enthalten sein:

Element	Beschreibung
Id	UUID des Projekts, in dem sich die Configuration `.wac` befindet.

10.21.12.1.1.2 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration>
    <General  displayName="My Web App" enableOverlayBlurredBackground="true"/>
    <Access publicAccess="false" />
    <Status active="true" localOnly="false" simpleDataTransport="true" />
</Configuration>
```

Das Element `<Configuration>` kann folgende Elemente enthalten:

Mögliche Elemente

Element	Beschreibung
`General`	Enthält Konfigurationsdefinitionen zu der allgemeinen Darstellung der Web App.
`Access`	Enthält Konfigurationsdefinitionen zum Zugriff auf die Web App.
`Status`	Enthält Konfigurationsdefinitionen zu den internen Statuswerten der Web App.
`Logout`	Enthält Konfigurationsdefinitionen zum Logout.

Weitere Informationen finden Sie im Abschnitt

- Operation: Get Configuration by Id
- Operation: List WebApp Configurations
- Operation: Update Configuration

10.21.12.2 Operation: List WebApp Configurations

Die Operation `List WebApp Configurations` listet alle Configurations im Repository auf.

10.21.12.2.1 Beispiele

10.21.12.2.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<Configurations>
   <Configuration>
      <Id>0cbff222-12e8-43f3-a2b8-31388c50afee</Id>
      <RepositoryName>1</RepositoryName>
      <ProjectName>MyWebAppProject</ProjectName>
   </Configuration>
</Configurations>
```

Folgende Elemente sind im Output enthalten:

Element	Beschreibung
Id	UUID des Projekts, in dem sich die Configuration `.wac` befindet.
RepositoryName	Name des X4 Repository-Ordners.
ProjectName	Name des Projekts im X4 Repository.

Weitere Informationen finden Sie im Abschnitt

- Operation: Get Configuration by Id
- Operation: List WebApp Configurations
- Operation: Update Configuration

10.21.12.3 Operation: Update Configuration

Die Operation `Update Configuration` aktualisiert die Configuration, die sich im Projekt mit der angegebenen ID befindet.

Hinweis:
Die Änderungen werden nur zur Laufzeit aktualisiert. Die `.wac`-Datei im Repository wird nicht aktualisiert.
Diese Attribute können nicht zur Laufzeit aktualisiert werden:

- `<Notifications />`
- `<HeaderIncludes />`

Beispiele

10.21.12.3.1 Input

Der Adapter erwartet ein XML-Dokument (siehe Beispiel) als Input.

```
<?xml version="1.0" encoding="UTF-8"?>
<Update>
    <Id>d84cf575-b071-42f4-8351-8436e87da8d1</Id>
    <Configuration>
        <Url path="WebAppProject_Controls" />
        <General displayName="My New Web App"
            enableCookieConsent="true" enableOverlayBlurredBackground="true"
            faviconUrl="favicon.png" logoUrl="icon:3d_rotation" logoColor="50"
            imprintLink="Imprint.de" dprLink="DPR.de" enableUrlParameters="true"
            showFooter="true" showHeader="true" />
        <Access accessRight="Admin" authorizationFlow="Public Access"
            realm="X4Realm" client="X4" clientSecret="A8M5pezJvqp9G)Nfx6aw9A((" />
        <Logout redirectUrl="https://www.softproject.de/" />
        <Status active="true" localOnly="true" />
        <MapAPIKey></MapAPIKey>
        <HTTPSecurityHeaders XFrameOptions="SAMEORIGIN">
            <StrictTransportSecurity enabled="false"
                preLoad="false" includeSubDomains="false" maxAge="86400" />
        </HTTPSecurityHeaders>
    </Configuration>
</Update>
```

Folgende Elemente müssen im Input enthalten sein:

Element	Beschreibung
`Update`	Das Wurzelelement des XML-Dokuments.
`Id`	UUID des Projekts, in dem sich die Configuration `.wac` befindet.
`Configuration`	Das Wurzelelement der `.wac`-Datei
`SecurityHTTPHeaders`	Wird verwendet, um die Web App vor unerwünschten Angriffen zu schützen.

Die Attribute der Elemente finden Sie im Web App-Handbuch Bereich Configuration deklarieren.

Weitere Informationen:

- Operation: Get Configuration by Id
- Operation: List WebApp Configurations
- Operation: Update Configuration

10.21.13 Web App Theme Management

`Web App Theme Management`: Aktualisiert oder listet Web App Theme-Dateien auf, die sich im X4 Repository oder in einem Projekt befinden.

Weitere Informationen zum Theming (Farben und Schriftarten) finden Sie im **Web App-Handbuch**.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `List WebApp Themes`: Liste mit allen Web App Theme im Repository ausgeben • `Get Theme by Id`: Web App Theme mit der angegebenen Id ausgeben • `Update Theme`: Web App Theme mit der angegebenen Id aktualisieren

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.server.webapp.adapter.WebAppThemeConfigurationAdapter*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Weitere Informationen:

- Operation: Get Theme by Id
- Operation: List WebApp Themes
- Operation: Update Theme

10.21.13.1 Operation: Update Theme

Die Operation `Update Theme` aktualisiert das Theme, das sich im Projekt mit der angegeben Id befindet.

Die Änderungen werden nur zur Laufzeit aktualisiert. Die Theme-Datei vom Dateityp `.theme` im Repository wird nicht aktualisiert.

Weitere Informationen zum Theming (Farben und Schriftarten) finden Sie im **Web App-Handbuch**.

10.21.13.1.1 Beispiele

10.21.13.1.1.1 Input

Der Adapter erwartet ein XML-Dokument als Input.

```
<?xml version='1.0' encoding='UTF-8'?>
<Update>
    <Id>a902c4c7-99a1-40e7-8c8d-16fbaec8ad7b</Id>
    <Theme>
        <ColorScheme>
            <Color50 normal="#ffffff" contrast="#000000" />
            <Color100 normal="#f8f8f8" contrast="#000000" />
            <Color200 normal="#e6e9f2" contrast="#000000" />
            <Color300 normal="#f0f0f0" contrast="#000000" />
            <Color400 normal="#43548e" contrast="#ffffff" />
            <Color500 normal="#1e3278" contrast="#ffffff" />
            <Color600 normal="#cbcbcb" contrast="#000000" />
            <Color700 normal="#aeaeae" contrast="#ffffff" />
            <Color800 normal="#919191" contrast="#ffffff" />
            <Color900 normal="#2b2b2b" contrast="#ffffff" />
            <ColorA100 normal="#f3f3f3" contrast="#000000" />
            <ColorA200 normal="#7180b5" contrast="#ffffff" />
            <ColorA400 normal="#ececec" contrast="#000000" />
            <ColorA700 normal="#ff5a00" contrast="#ffffff" />
        </ColorScheme>
        <ChartScheme>
            <Color01 normal="#e7aada" contrast="#000000" />
            <Color02 normal="#84c08c" contrast="#000000" />
            <Color03 normal="#adb7f4" contrast="#000000" />
            <Color04 normal="#d4e2a2" contrast="#000000" />
            <Color05 normal="#4abfe8" contrast="#000000" />
            <Color06 normal="#eabf8d" contrast="#000000" />
            <Color07 normal="#78ccf9" contrast="#000000" />
            <Color08 normal="#abb87a" contrast="#000000" />
            <Color09 normal="#f6a2ac" contrast="#000000" />
            <Color10 normal="#82e8db" contrast="#000000" />
            <Color11 normal="#9cdfb4" contrast="#000000" />
            <Color12 normal="#55cdd8" contrast="#000000" />
            <ColorGaugeAxisMin normal="#228b22" contrast="#ffffff" />
            <ColorGaugeAxisMiddle normal="#3e5570" contrast="#ffffff" />
            <ColorGaugeAxisMax normal="#ff4500" contrast="#ffffff" />
        </ChartScheme>
        <Fonts>
            <MainFont>
                <Font sourceType="web">Arial</Font>
            </MainFont>
            <Font01>
                <Font sourceType="system">Calibri</Font>
                <Font sourceType="web">https://fonts.google.com/specimen/Noto+Sans</
Font>
            </Font01>
            <Font02>
                <Font sourceType="web">https://fonts.google.com/specimen/
Lora#standard-styles</Font>
            </Font02>
        </Fonts>
    </Theme>
</Update>
```

Folgende Elemente müssen im Input enthalten sein:

Element	Beschreibung
`Update`	Das Wurzelelement des XML-Dokuments.
`Id`	UUID des Projekts, in dem sich die Theme-Datei vom Dateityp `.theme` befindet.
`Theme`	Das Wurzelelement der Theme-Datei.

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters ist ein Fehler aufgetreten.

Weitere Informationen:

- Operation: Update Theme
- Operation: Get Theme by Id
- Operation: List WebApp Themes

10.21.13.2 Operation: Get Theme by Id

Die Operation `Get Theme by Id` gibt die Theme-Datei aus, die sich im Projekt mit der angegeben Id befindet.

Wenn Änderungen zur Laufzeit vorgenommen wurden, wird der zur Laufzeit verwendete Inhalt der Theme-Datei ausgegeben.

Weitere Informationen zum Theming (Farben und Schriftarten) finden Sie im **Web App-Handbuch**.

10.21.13.2.1 Beispiele

10.21.13.2.1.1 Input

Der Adapter erwartet ein XML-Dokument als Input:

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>fa2c23b6-1843-4758-b686-06f3f748afff</Id>
```

Folgende Elemente müssen im Input enthalten sein:

Element	Beschreibung
`Id`	UUID des Projekts, in dem sich die Theme-Datei mit dem Dateityp `.theme` befindet.

10.21.13.2.1.2 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<Theme>
   <ColorScheme>
      <Color50 contrast="#000000" normal="#ffffff"/>
      <Color100 contrast="#000000" normal="#f8f8f8"/>
      <Color200 contrast="#000000" normal="#e6e9f2"/>
      <Color300 contrast="#000000" normal="#f0f0f0"/>
      <Color400 contrast="#ffffff" normal="#ececec"/>
      <Color500 contrast="#ffffff" normal="#1e3278"/>
      <Color600 contrast="#000000" normal="#cbcbcb"/>
      <Color700 contrast="#ffffff" normal="#aeaeae"/>
      <Color800 contrast="#ffffff" normal="#919191"/>
      <Color900 contrast="#ffffff" normal="#2b2b2b"/>
      <ColorA100 contrast="#000000" normal="#f3f3f3"/>
      <ColorA200 contrast="#ffffff" normal="#7180b5"/>
      <ColorA400 contrast="#000000" normal="#ececec"/>
      <ColorA700 contrast="#ffffff" normal="#ff5a00"/>
   </ColorScheme>
   <ChartScheme>
      <Color01 contrast="#000000" normal="#e7aada"/>
      <Color02 contrast="#000000" normal="#84c08c"/>
      <Color03 contrast="#000000" normal="#adb7f4"/>
      <Color04 contrast="#000000" normal="#d4e2a2"/>
      <Color05 contrast="#000000" normal="#4abfe8"/>
      <Color06 contrast="#000000" normal="#eabf8d"/>
      <Color07 contrast="#000000" normal="#78ccf9"/>
      <Color08 contrast="#000000" normal="#abb87a"/>
      <Color09 contrast="#000000" normal="#f6a2ac"/>
      <Color10 contrast="#000000" normal="#82e8db"/>
      <Color11 contrast="#000000" normal="#9cdfb4"/>
      <Color12 contrast="#000000" normal="#55cdd8"/>
      <ColorGaugeAxisMin contrast="#ffffff" normal="#228b22"/>
      <ColorGaugeAxisMiddle contrast="#ffffff" normal="#3e5570"/>
      <ColorGaugeAxisMax contrast="#ffffff" normal="#ff4500"/>
   </ChartScheme>
   <Fonts>
        <MainFont>
            <Font sourceType="system">Arial</Font>
        </MainFont>
        <Font01>
            <Font sourceType="system">Calibri</Font>
            <Font sourceType="web">https://fonts.google.com/specimen/Noto+Sans</Font>
        </Font01>
        <Font02>
            <Font sourceType="web">https://fonts.google.com/specimen/Lora#standard-
styles</Font>
        </Font02>
    </Fonts>
</Theme>
```

Das Element <Theme> kann folgende Elemente enthalten:

Mögliche Elemente

Element	Beschreibung
`ColorScheme`	Enthält Definitionen des Main Themes, das in der Web App verwendet wird.
`ChartScheme`	Enthält Definitionen des Chart Themes, das in der Web App verwendet wird.
`Fonts`	Enthält Definitionen der Schriftarten, die in der Web App verwendet werden.

Weitere Informationen:

- Operation: Update Theme
- Operation: Get Theme by Id
- Operation: List WebApp Themes

10.21.13.3 Operation: List WebApp Themes

Die Operation `List WebApp Themes` listet alle Projekte mit einer Theme-Datei im Repository auf.

Weitere Informationen zum Theming (Farben und Schriftarten) finden Sie im **Web App-Handbuch**.

10.21.13.3.1 Beispiele

10.21.13.3.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<Themes>
   <Theme>
      <Id>a902c4c7-99a1-40e7-8c8d-16fbaec8ad7b</Id>
      <RepositoryName>1</RepositoryName>
      <ProjectName>MyWebAppProject_1</ProjectName>
   </Theme>
   <Theme>
      <Id>9313feeb-429a-4249-927b-ac51c230d198</Id>
      <RepositoryName>1</RepositoryName>
      <ProjectName>MyWebAppProject_2</ProjectName>
   </Theme>
</Themes>
```

Folgende Elemente sind im Output enthalten:

Element	Beschreibung
`Id`	UUID des Projekts, in dem sich die Configuration `.wac` befindet.
`RepositoryName`	Name des X4 Repository-Ordners.
`ProjectName`	Name des Projekts im X4 Repository.

Weitere Informationen:

- Operation: Update Theme
- Operation: Get Theme by Id
- Operation: List WebApp Themes

10.22 X4 Tools

In der Kategorie X4 Tools werden alle Adapter gebündelt, die Funktionalitäten der X4 BPMS zur Verwendung in technischen Prozessen bereitstellen:

- Case Management Documents
- Case Management
- Case Management Statistics
- Commandline
- Document Comparison
- Document Properties Changer
- Empty Output Adapter
- Global Context Manipulation
- Keycloak Token
- Log4J Logging
- Loop Adapter
- Parallel Process Starter
- Process Starter
- Projects
- Project Access
- System Status
- Task Management
- User Identity Adapter
- User Session Adapter
- Web App Translation
- X4 Management
- X4 Server Config Reader
- X4 Variable Collector

10.22.1 Case Management Documents

`Case Management Documents`: Bildet die Schnittstelle zur „Case Management"-Funktionalität der Produktkomponente X4 BPM. Der Adapter ermöglicht das Verwalten von Dokumenten (Dateien) an Vorgängen. Dokumente können zusätzlich zu Merkmalen des Fachdatenmodells direkt an Vorgängen hinterlegt werden.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `AttachDocument`: Fügt dem Vorgang ein Dokument hinzu. • `UpdateDocument`: Aktualisiert ein vorhandenes Dokument. • `UpdateDocumentMetadata`: Aktualisiert die Metadaten eines vorhandenen Dokuments. • `GetDocument`: Gibt das Dokument als Output aus. • `GetDocumentMetadata`: Gibt die Metadaten eines Dokuments als Output aus. • `ListAllDocumentMetadata`: Gibt die Metadaten aller Dokumente des Vorgangs als Output aus. • `RemoveDocument`: Entfernt das Dokument aus dem Vorgang.

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.server.bmn.casemanagement.adapter.CaseDocumentAdapter*: Hauptklasse (Standard)
`caseId`	ID des zu verarbeitenden Vorgangs **Mögliche Werte:** • Beliebige (gültige) Case-ID • `%CASE_ID%`: aktuelle Case-ID (Standard)
`title`	Titel des Dokuments **Mögliche Werte:** Beliebige Zeichenkette ⚠ Der Titel des Dokuments dient zur Identifizierung und muss innerhalb eines Prozesses eindeutig sein!
`fileName`	Dateiname des Dokuments **Mögliche Werte:** Beliebige Zeichenkette, z. B. `Anhang/Rechnung.pdf`
`encoding`	Zeichenkodierung des Dokuments **Mögliche Werte:** Gültige Angabe der Zeichenkodierung, z. B. `UTF-8`
`mimeType`	MIME-Type des Dokuments **Mögliche Werte:** Gültige Angabe des MIME-Types, z. B. `application/xml`

`base64`	Gibt an, ob das Dokument als Base64-Zeichenkette übergeben wird. **Mögliche Werte:** • `true`: Dokument wird als Base64-Zeichenkette übergeben. • `false`: Dokument wird direkt übergeben (Standard).

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation konnte nicht ausgeführt werden.

Input

Der Adapter erwartet keinen speziellen Input.

> ⓘ Wenn bei den Operationen `AttachDocument` und `UpdateDocument` der Parameter `base64` auf *`true`* gesetzt ist, muss der Input eine Base64-kodierte Zeichenkette sein.

Output:

Je nach Operation gibt der Adapter Daten in verschiedenen XML-Strukturen aus:

- Bei folgenden Operationen wird der Input bei einer erfolgreichen Operation wieder als Output ausgegeben:
  - `AttachDocument`
  - `UpdateDocument`
  - `UpdateDocumentMetadata`
  - `RemoveDocument`
- `GetDocumentMetadata`-Operation

  Beispiel-Output für die Operation GetDocumentMetadata

  ```
  <?xml version="1.0" encoding="UTF-8" standalone="true"?>
  <Document updatedOn="2019-03-28T13:34:46.233+01:00" updatedBy="x4" createdOn="2
  019-03-28T13:34:45.690+01:00" createdBy="x4">
      <Title>Rechnung</Title>
      <FileName>Rechnung.xml</FileName>
      <MimeType>application/xml</MimeType>
      <Encoding>UTF-8</Encoding>
  </Document>
  ```

- `ListAllDocumentMetadata`-Operation

Beispiel-Output für die Operation ListAllDocumentMetadata

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<Documents>
    <Document createdOn="2019-03-28T13:35:04.543+01:00" createdBy="x4">
        <Title>Kommentar</Title>
        <FileName>Kommentar.txt</FileName>
        <MimeType>text/plain</MimeType>
        <Encoding>UTF-8</Encoding>
    </Document>
    <Document createdOn="2019-03-28T13:34:45.690+01:00" createdBy="x4"
updatedOn="2019-03-28T13:34:46.233+01:00" updatedBy="x4">
    <Title>Rechnung</Title>
        <FileName>Rechnung.xml</FileName>
        <MimeType>application/xml</MimeType>
        <Encoding>UTF-8</Encoding>
    </Document>
</Documents>
```

- `GetDocument`-Operation:
  Wenn der Parameter `base64` auf *`true`* gesetzt ist, wird das Dokument als Base64-kodierte Zeichenkette ausgegeben.

10.22.2 Case Management

`Case Management`: Bildet die Schnittstelle zur „Case Management"-Funktionalität der Produktkomponente X4 BPM. Der Adapter ermöglicht beispielsweise das Setzen und Auslesen von Vorgangsdaten wie Fachstatus oder Merkmalen des Fachdatenmodells.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `SetFeature`: Schreibt ein Feature • `SetFeatures`: Schreibt mehrere Features in einem Vorgang • `GetFeatures`: Gibt alle Features des Vorgangs zurück • `SetState`: Setzt den Status des Vorgangs • `GetStates`: Gibt alle möglichen Status des Vorgangs zurück • `GetComments`: Gibt alle Kommentare des Vorgangs zurück • `GetInformation`: Gibt alle Informationen des Vorgangs zurück

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.server.bmn.casemanagement.adapter.CaseManagementAdapter*: Hauptklasse (Standard)

`CaseId`	ID des zu verarbeitenden Vorgangs **Mögliche Werte:** • Beliebige gültige Benutzer-ID • *`%CASE_ID%`*: Aktuelle Case-ID (Standard)
`CaseProcess`	Repository-Pfad zu einem Business Process, in dem der entsprechende Vorgang definiert ist. **Mögliche Werte:** • Beliebiger kompletter Repository-Pfad, z. B. *`Project/Folder/Process.bpm`* • Beliebiger gültiger Platzhalter, z. B. *`%START_PROCESS%`* • *`%START_PROCESS%`*: Aufrufender Business Process (Standard)
`CaseUserId`	Benutzer-ID des Besitzers eines Vorgangs **Mögliche Werte:** • Beliebige gültige Benutzer-ID • *`%USER_ID%`*: Aktuelle Benutzer-ID (Standard)
`dataformatVersion`	Version des verwendeten Datenformats **Mögliche Werte:** • `1.0`: Verwendetes Datenformat bis Version 7.0 • `2.0`: Verwendetes Datenformat ab Version 7.0, siehe Input-Beispiele

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde durchgeführt, es wurde aber kein Ergebnis gefunden oder keine Daten geändert.
`-1`	Die Operation konnte nicht ausgeführt werden.

Input

Der Adapter erwartet eine vordefinierte XML-Struktur als Input. Die Struktur ist abhängig von der verwendeten Operation.

Wird bei setFeature / setFeatures das Attribut "type" nicht angegeben, ist der Datentyp "String". Die anderen möglichen Datentypen der "Case Management"-Funktionalität entsprechen denen der dortigen Typen.

- `SetFeature`-Operation:

Erwartete Struktur für die Operation SetFeature

```
<?xml version="1.0" encoding="UTF-8" ?>
<Name>Max</Name>
```

- `SetFeatures`-Operation:

Erwartete Struktur für die Operation SetFeatures

```
<?xml version="1.0" encoding="UTF-8" ?>
<Features>
    <Name type="String">Max</Name>
    <Lastname type="String">Mustermann</Lastname>
    <Date type="DateTime">2018-09-27T00:00:00.000+00:00</Date>
    <IsCustomer type="Boolean">true</IsCustomer>
    <Amount type="Decimal">111.25</Amount>
    <InternalNumber type="Integer">45</InternalNumber>
</Features>
```

- `SetState`-Operation:

Erwartete Struktur für die Operation SetState

```
<?xml version="1.0" encoding="UTF-8" ?>
<State>NEW</State>
```

Output:

Je nach Operation gibt der Adapter Daten in verschiedenen XML-Strukturen aus:

- `GetFeatures`-Operation

Beispiel-Output für die Operation GetFeatures

```
<?xml version="1.0" encoding="UTF-8" ?>
<Features>
    <Name>Max</Name>
    <Lastname>Mustermann</Lastname>
    <Address>Musterweg 1</Address>
</Features>
```

- `GetStates`-Operation

Beispiel-Output für die Operation GetStates

```
<?xml version="1.0" encoding="UTF-8" ?>
<CaseStates>
    <State>NEW</State>
    <State>DONE</State>
</CaseStates>
```

- `GetInformation`-Operation

Beispiel-Output für die Operation GetInformation

```
<?xml version="1.0" encoding="UTF-8" ?>
<Information>
    <Case creator="x4" created="2018-12-14T16:39:48.100+01:00">
        <Comments />
        <Features>
            <Name>Max</Name>
            <Lastname>Mustermann</Lastname>
            <Address>Musterweg 1</Address>
        </Features>
        <State>NEW</State>
    </Case>
    <PossibleCaseStates>
        <State>NEW</State>
        <State>DONE</State>
    </PossibleCaseStates>
</Information>
```

10.22.3 Case Management Statistics

`Case Management Statistics`: Stellt Statistiken für Vorgänge (Business Process) und Human Tasks bereit.

Eigenschaften

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • `GetAllCaseMetadata`: Alle Case-Metadaten bereitstellen, die dem Filterkriterium entsprechen • `GetAllTaskMetadata`: Alle Task-Metadaten bereitstellen, die dem Filterkriterium entsprechen

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.server.bpm.casemanagement.adapter.CaseManagementStatisticsAdapter* Hauptklasse (Standard)

Statuswerte

1	Die Operation wurde erfolgreich ausgeführt.
-1	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Je nach Operation erwartet der Adapter ein spezielles XML-Dokument als Input.

Operation GetAllCaseMetadata

```
<?xml version="1.0" encoding="UTF-8"?>
<CaseMetadataFilter>
    <!-- Filtert alle Cases nach dem angegebenen Startdatum -->
    <From>2019-10-01T08:00:00.000</From>
    <!-- Filtert alle Cases nach dem angegebenen Enddatum -->
    <To>2019-10-07T08:00:00.000</To>
    <!-- Filtert alle Cases nach dem angegebenen Prozessnamen -->
    <ProcessName>Invoice/Processes/InvoiceApproval.bpm</ProcessName>
    <!-- Filtert alle Cases nach dem angegebenen Case Status -->
    <CaseState>Approved</CaseState>
    <!-- Filtert alle Cases nach dem angegebenen technischen Status
        mögliche Werte: RUNNING, CRASHED, WAITING, FINISHED, FAILED, UNKNOWN -->
    <TechnicalState>FINISHED</TechnicalState>
</CaseMetadataFilter>
```

Operation GetAllTaskMetadata

```
<?xml version="1.0" encoding="UTF-8"?>
<TaskMetadataFilter>
    <!-- Filtert alle Human Tasks nach dem angegebenen Startdatum -->
    <From>2019-10-01T08:00:00.000</From>
    <!-- Filtert alle Human Tasks nach dem angegebenen Enddatum -->
    <To>2019-10-07T08:00:00.000</To>
    <!-- Filtert alle Human Tasks nach dem angegebenen Prozessnamen -->
    <ProcessName>Invoice/Processes/InvoiceApproval.bpm</ProcessName>
    <!-- Filtert alle Human Tasks nach dem angegebenen Task Status
    mögliche Werte: OPEN, CLOSED, IN_PROGRESS, DEADLINE_REACHED -->
    <TaskState>OPEN</TaskState>
    <!-- Filtert alle Human Tasks nach dem angegebenen Case Status -->
    <CaseState>To be approved by clerk</CaseState>
    <!-- Filtert alle Human Tasks nach der angegebenen Frist-->
    <Deadline>2019-10-07T08:00:00.000</Deadline>
    <!-- Filtert alle Human Tasks nach dem angegebenen Empfänger (User oder Gruppe
(Angabe des technischen Bezeichners) ) -->
    <Assignment type="GROUP">clerks</Assignment>
    <!-- Filtert alle Human Tasks nach dem angegebenen Besitzer -->
    <Owner>emil</Owner>
</TaskMetadataFilter>
```

⚠ Beachten Sie, dass beim Filtern anhand des Empfängers für Gruppen, d.h. `Assignment` des Typs *GROUP*,der technische Bezeichner und nicht der Name der Gruppe angegeben werden muss.

Ausgabe

Je nach Operation liefert der Adapter ein spezielles XML-Dokument als Output.

Operation GetAllCaseMetadata

```
<?xml version="1.0" encoding="UTF-8"?>
<CaseManagementStatistics>
    <CaseMetadata caseId="86757d17-5a8c-47c8-84b1-3fc9f241bf9f">
        <CaseState>Approved</CaseState>
        <ProcessName>Invoice/Processes/InvoiceApproval.bpm</ProcessName>
        <StartDate>2019-10-02T09:27:47.432+02:00</StartDate>
        <EndDate>2019-10-02T09:27:47.695+02:00</EndDate>
        <Duration>263</Duration>
        <TechnicalState>FINISHED</TechnicalState>
    </CaseMetadata>
    <CaseMetadata caseId="a4f12abd-97ad-48cb-a969-40ad84e53ecd">
        <CaseState>Approved</CaseState>
        <ProcessName>Invoice/Processes/InvoiceApproval.bpm</ProcessName>
        <StartDate>2019-10-04T14:27:47.146+02:00</StartDate>
        <EndDate>2019-10-07T14:27:47.842+02:00</EndDate>
        <Duration>696</Duration>
        <TechnicalState>FINISHED</TechnicalState>
    </CaseMetadata>
    ...
</CaseManagementStatistics>
```

Operation GetAllTaskMetadata

```
<?xml version="1.0" encoding="UTF-8"?>
<CaseManagementStatistics>
    <TaskMetadata taskId="fa24c1d0-e9f0-4a72-bca0-760c1cb1d653">
        <CaseReference>846fac61-ff14-b349-6452-12afb6487ac5</CaseReference>
        <HumanTaskReference>aeee7252-fcc2-4da8-b958-6e16dbaecadf</HumanTaskReference>
        <Title>Protokoll hochladen</Title>
        <ProcessName>Invoice/Processes/InvoiceApproval.bpm</ProcessName>
        <TaskState>OPEN</TaskState>
        <CaseState>To be approved by clerk</CaseState>
        <Deadline>2019-10-05T16:00:00.000+02:00</Deadline>
        <Assignment type="GROUP">clerks</Assignment>
        <Owner>emil</Owner>
        <StartDate>2019-10-01T08:14:41.395+02:00</StartDate>
        <EndDate>2019-10-02T10:48:12.753+02:00</EndDate>
    </TaskMetadata>
    <TaskMetadata taskId="9b330bc3-310a-4638-ac1b-07f115d564d7">
        <CaseReference>946ac5f6-64a8-3649-accf-69acf5a96fc5</CaseReference>
        <HumanTaskReference>5f377252-faa8-12fa-b958-6e16dbaecadf</HumanTaskReference>
        <Title>Rechnung freigeben</Title>
        <ProcessName>Invoice/Processes/InvoiceApproval.bpm</ProcessName>
        <TaskState>OPEN</TaskState>
        <CaseState>To be approved by clerk</CaseState>
        <Deadline>2019-10-06T16:00:00.000+02:00</Deadline>
        <Assignment type="USER">clerks</Assignment>
        <Owner>emil</Owner>
        <StartDate>2019-10-01T13:27:53.593+02:00</StartDate>
        <EndDate>2019-10-04T11:56:09.852+02:00</EndDate>
    </TaskMetadata>
    ...
</CaseManagementStatistics>
```

10.22.4 Commandline

`Commandline`: Ruft eine externe Anwendung oder einen externen Befehl über eine Kommandozeile des Betriebssystems mit beliebig vielen Argumenten auf. Der Adapter erwartet eine adapterspezifische Input-XML-Struktur, die Base64-kodierte Nutzdaten enthalten kann und gibt ein XML-Dokument mit den Base64-kodierten Ergebnissen des Aufrufs von *StdOut* und/oder *StdErr* sowie dem zurückgegebenen *Exit Code* zurück.
Ehemalige Bezeichnung: *OS Process Execution Adapter*

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt. **Mögliche Werte:** *`Execute`*: Anwendung/Befehl über eine Kommandozeile aufrufen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.osprocess.OsProcessAdapter*: Hauptklasse (Standard)
WorkingDirectory	Arbeitsverzeichnis für den ausgeführten Anwendungsaufruf; wird ggf. durch das im Input-XML gesetzte Arbeitsverzeichnis überschrieben; falls beides leer ist, wird das *X4 Server*-Arbeitsverzeichnis verwendet **Mögliche Werte:** • Beliebige gültige Pfadangabe (z.B. *C:\X4\temp*) • *(leer)*: Kein Standard-Arbeitsverzeichnis (über Input-XML manuell setzen oder das *X4 Server*-Arbeitsverzeichnis wird verwendet) (Standard)
CollectStdOut	*Standard Output* -Strom des aufgerufenen Programms erfassen **Mögliche Werte:** • *true*: *Standard Output*-Strom des aufgerufenen Programms im Ergebnis-XML-Dokument in Element <StdOut> Base64-kodiert speichern (Standard) • *false*: *Standard Output* nicht berücksichtigen (Element <StdOut> im Ergebnis-XML-Dokument bleibt leer)
CollectStdErr	*Standard Error* -Strom des aufgerufenen Programms erfassen **Mögliche Werte:** • *true*: *Standard Error*-Strom des aufgerufenen Programms im Ergebnis-XML-Dokument in Element <StdErr> Base64-kodiert speichern (Standard) • *false*: *Standard Error* nicht berücksichtigen (Element <StdErr> im Ergebnis-XML-Dokument bleibt leer)
MergeStdErrIntoStdOut	*Standard Error*-Strom zusammen mit dem *Standard Output* -Strom in Element <StdOut> Base64-kodiert ausgeben **Mögliche Werte:** • *true*: *Standard Error* und *Standard Output* zusammen ausgeben • *false*: Datenströme separat ausgeben (Standard)

Input

Der Adapter erwartet ein XML-Dokument, das nach folgendem Muster aufgebaut ist:

```
<Command>
  <WorkingDirectory><!--Arbeitsverzeichnis (optional)--></WorkingDirectory>
  <Application><!--Name der Kommandozeile oder kompletter Pfad der Anwendung--></
Application>
  <!-- Beliebig viele Argumente (Reihenfolge beachten!) -->
  <Argument><!--Argument/Parameter--></Argument>
  <Environment clear="nur mit folgenden Umgebungsvariablen arbeiten (true/false)">
     <!-- beliebig viele Umgebungsvariablen -->
     <Entry name="Umgebungsvariablenname"><!--Wert--></Entry>
  </Environment>
  <Input><!--Base64-kodierte Daten zur Verarbeitung durch die externe Anwendung--></
Input>
  <Copy><!--Beliebige Zeichenkette/XML-Struktur zum unveränderten Durchschleifen--></
Copy>
</Command>
```

Erläuterung:

- `<WorkingDirectory>` (optional): Arbeitsverzeichnis in dem die aufgerufene Anwendung operiert; überschreibt den Wert in Parameter `WorkingDirectory`); entspricht *nicht* dem Pfad der Anwendung (siehe unten)!
- `<Application>`: Name der Kommandozeile oder Anwendungsname mit oder ohne komplettem Pfad (falls der Anwendungspfad bereits in den Umgebungsvariablen verfügbar ist)
  Alternativ unter Windows `cmd.exe` verwenden und die Anwendung als erstes Argument angeben
- `<Environment>`: Definiert Umgebungsvariablen; Über das Attribut `clear="true"` können Sie steuern, ob der Befehl nur mit den Umgebungsvariablen, die innerhalb von Element `<Environment>` definiert wurden, ausgeführt wird. Mit `clear="false"` werden die innerhalb von Element `<Environment>` definierten Umgebungsvariablen zu den bestehenden hinzugefügt.
- `<Input>` (optional) Enthält Base64-kodierte Input-Daten
- `<Copy>` (optional): Enthält eine XML-Struktur/beliebige Zeichenkette zum unveränderten Durchschleifen

> Im Gegensatz zu den Adapter-Parametern können Sie im Input-XML keine Platzhalter wie z.B. `%USER_COLLECTION%` verwenden!

Beispiel

In der folgenden Input-XML-Struktur wird die Standardkommandozeile von Windows, `cmd.exe`, im Arbeitsverzeichnis `C:\` aufgerufen und es werden mit dem Befehlsargument `set` die Umgebungsvariablen `name` und `age` gesetzt. Über das Argument `/c` wird dabei Befehl in der Zeichenfolge ausgeführt und dann beendet. Über `clear="true"` wird der Befehl nur mit den hier definierten Umgebungsvariablen ausgeführt.

```
<Command>
  <WorkingDirectory>C:\</WorkingDirectory>
  <Application>cmd.exe</Application>
  <Argument>/c</Argument>
  <Argument>set</Argument>
  <Environment clear="true">
    <Entry name="name">Theo Tester</Entry>
    <Entry name="age">35</Entry>
  </Environment>
  <Input></Input>
  <Copy></Copy>
</Command>
```

Ausgabe

Der Funktions-Adapter gibt ein XML-Dokument aus, das nach folgendem Muster aufgebaut ist.

```
<CommandResult>
   <ExitCode>ExitCode der Anwendung</ExitCode>
   <StdOut>Base64-kodierte Ausgabedaten der Anwendung</StdOut>
   <StdErr>Base64-kodierte Fehlerausgabe der Anwendung</StdErr>
   <Copy>Unverändert durchgeschleifte Zeichenkette/XML-Struktur</Copy>
</CommandResult>
```

Statuswerte

1	Aufruf wurde ausgeführt; der Exitcode des Programms ist 0.
0	Aufruf wurde ausgeführt; der Exitcode des Programms ist ungleich 0.
-1	Aufruf konnte nicht ausgeführt werden (z. B. weil Programm nicht gefunden wurde).

10.22.5 Document Comparison

`Document Comparison`: Vergleicht eine Repository-Datei mit einer Input-Datei und gibt als Prozess-Ergebnis einen Statuswert aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`Compare`*: Dateien miteinander vergleichen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.comparison.X4ComparisonAdapter*: Hauptklasse (Standard)
comparisonMethod	Vergleichsmethode **Mögliche Werte:** • `BYTE`: Dateien byte-weise miteinander vergleichen (Standard) • `ZIP`: ZIP-Dateien anhand ihrer Inhalte miteinander vergleichen • `XML`: XML-Dateien anhand ihrer Inhalte miteinander vergleichen • `TEXT`: Text-Dateien anhand ihrer Inhalte miteinander vergleichen • `EMPTY`: Input-Dateien mit leeren Byte-Arrays vergleichen
expectedResult	Pfad zum Dokument das mit dem Adapter-Input verglichen werden soll. Bezieht sich auf die Vergleichsmethoden `BYTE`, `ZIP`, `XML` und `TEXT` **Mögliche Werte:** Repository-Pfad, z. B. *`X4Experience/BPM/1-Examples/InvoiceReceipt/BPM/Data/Invoice.xml`* (Pflichtfeld)
ignoreWhitespace	Gibt an, ob Leerzeichen beim Vergleichen ignoriert werden sollen; Bezieht sich auf die Vergleichsmethoden `XML` und `TEXT` **Mögliche Werte:** • `ja`: Leerzeichen ignorieren • `nein`: Leerzeichen berücksichtigen (Standard)
normalizeWhitespace	Gibt an, ob Leerzeichen am Anfang und Ende einer Zeichenkette entfernt werden sollen; Bezieht sich auf die Vergleichsmethoden `XML` und `TEXT` **Mögliche Werte:** • `ja`: Leerzeichen wird am Anfang und Ende einer Zeichenkette entfernt (Standard) • `nein`: Leerzeichen am Anfang und Ende einer Zeichenkette bleibt erhalten

ignoreComments	Gibt an, ob Kommentare innerhalb des XML-Dokuments beim Vergleichen ignoriert werden sollen; Bezieht sich auf die Vergleichsmethode `XML` **Mögliche Werte:** • `ja`: Kommentare ignorieren • `nein`: Kommentare berücksichtigen (Standard)
compareSimilar	Gibt an, ob die Reihenfolge der Elemente und Attribute (Tags) innerhalb des XML-Dokuments beim Vergleichen ignoriert werden soll, z. B. wenn die Inhalte innerhalb des XML-Dokuments zwar gleich, aber ihre Reihenfolge unterschiedlich ist; Bezieht sich auf die Vergleichsmethode `XML` **Mögliche Werte:** • `ja`: Reihenfolge ignorieren • `nein`: Reihenfolge berücksichtigen (Standard)
ignoreCDataAndTextDiff	Gibt an, ob der Inhalt innerhalb des Elementes `CDATA` als Klartext interpretiert werden soll; Bezieht sich auf die Vergleichsmethode `XML` **Mögliche Werte:** • `ja`: Inhalt von `CDATA` als Klartext interpretieren (Standard) • `nein`: Inhalt von `CDATA` nicht als Klartext interpretieren

Input

Als Input erwartet der Adapter eine beliebige Datei, die mit einer gegebenen Repository-Datei verglichen wird.

Ausgabe

Als Prozess-Ergebnis können verschiedene Statuswerte ausgegeben werden:

`1`	Die verglichenen Dokumente sind gleich
`0`	Die verglichenen Dokumente sind nicht gleich
`-1`	Die Dokumente konnten nicht miteinander verglichen werden, da der Input entweder leer, nicht wohlgeformt oder falsch war.

10.22.6 Document Properties Changer

`Document Properties Changer`: Liest und verändert Dokumenteigenschaften eines im X4 ESB für den Prozessdaten-Austausch verwendeten sogenannten `X4Document`. Dabei werden lediglich die intern verwendeten Dokumenteigenschaften verändert, was Einfluss auf die Ausführung von einigen Adaptern (nicht aber von XSL-Mappings) haben kann. Die eigentlichen Daten des Dokuments bleiben unverändert.

Ehemalige Bezeichnung: *Document Properties Adapter*

Um die Zeichenkodierung der Dokumentdaten zu ändern, den Charset Changer verwenden. Um die Zeichen der Dokumentdaten in eine andere Zeichenkodierung zu konvertieren, den Encoding Converter verwenden.
Beachten Sie zudem: Änderungen des Zeichenkodierung mit dem *Document Properties Changer* haben keinen Effekt auf XSL-Transformationen. Wenn sich keine XML-Deklaration mit Angabe der Zeichenkodierung im Dokument befindet, dann wird *UTF-8* gemäß XML-Spezifikation als Standard-Zeichenkodierung verwendet.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *Change*: MIME-Typ und/oder Zeichenkodierung des `X4Document` ändern

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.documentproperties.DocumentPropertiesAdapter*: Hauptklasse (Standard)
`mimeType`	Neuer MIME-Typ des X4Document **Mögliche Werte:** Beliebiger MIME-Typ (z.B. *text/xml*)
`encoding`	Neue Zeichenkodierung des `X4Document` **Mögliche Werte:** Beliebige Zeichenkodierung (z.B. *ISO-8859-1*)
`removeEncoding`	Entfernt die Zeichencodierung des `X4Document` Wenn der Parameter `removeEncoding` aktiviert ist, muss der Parameter `encoding` leer sein.

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters ist ein Fehler aufgetreten.

10.22.7 Empty Output Adapter

`Empty Output Adapter` : Leert den Datenstrom im Technical Process und dient so als Datensenke, um einen wohldefinierten Zustand innerhalb des Technical Processes zu schaffen. Insbesondere im Produktionsmodus des *X4 Servers* ist dieser Adapter hilfreich, um eine leere Datei zu erzeugen, ohne

dass Zugriff auf ein Dateisystem erforderlich ist. Dieser verarbeitet beliebige Input-Daten und gibt ein 0 Byte langes Binärdokument oder nichts (Null) aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`Empty`*: Dokument mit 0 Byte Länge ausgeben • *`Null`*: Nichts (Null) ausgeben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.sink.Sink`*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.22.8 Global Context Manipulation

`Global Context Manipulation`: Bietet die Möglichkeit, ähnlich wie bei Prozess-Variablen beliebige Dokumente aus einem Technical Process in einer globalen Hashtabelle zwischenzuspeichern, sodass auch von anderen Prozessen auf diesen Kontext zugegriffen werden kann, siehe *Handbuch X4 Designer – Variablen-Baustein.*

Die im globalen Kontext abgelegten Dokumente werden unter einem Schlüssel gespeichert und sind über die gesamte Laufzeit des *X4 Servers* verfügbar. Der Funktions-Adapter kann Einträge in der Hash-Tabelle erstellen, überschreiben und löschen sowie alle oder ausgewählte Schlüssel der Hashtabellen-Einträge auflisten.

> Da ein Kontext *X4 Server*-weit gilt und alle Technical Processes auf dieselbe Hashtabelle zugreifen, ist es möglich, dass andere Threads parallel ein Dokument unter demselben Schlüssel einstellen. Zudem können geringe Zeitdifferenzen zwischen den Operationen `Put` und `PutGet` uneindeutige (z. B. leere) Ergebnisse liefern.
> Innerhalb von XSL-Mappings können Sie über das `globalcontext:`-URL-Protokoll zur Laufzeit auf bereits definierte Kontextschlüssel zugreifen, siehe *X4 Developer Guide – globalcontext:-URL-Protokoll*.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`Get`*: Eintrag, dessen Schlüssel in Parameter `Key` festgelegt wurde, aus dem globalen Kontext auslesen und das zuletzt vollständig gespeicherte Dokument ausgeben • *`Put`*: Input-Dokument mit dem in Parameter `Key` festgelegten Schlüssel im globalen Kontext speichern und liefert den gesetzten Eintrag zurück • *`PutGet`*: Input-Dokument mit einem Schlüssel im globalen Kontext speichern und das zuletzt vollständig gespeicherte Dokument ausgeben (dass das aktuellste Dokument ausgegeben wird, kann nicht garantiert werden!) • *`Remove`*: Eintrag, dessen Schlüssel in Parameter `Key` festgelegt wurde, aus dem Globalen Kontext löschen und gibt den gelöschten Eintrag zurück • *`List`*: Alle Schlüssel des globalen Kontexts in einem XML-Dokument ausgeben • *`RemoveAllMatching`*: Alle Einträge des globalen Kontexts löschen, deren Schlüssel dem Regex-Filterausdruck in Parameter `Filter` entsprechen • *`KeepAllMatching`*: Nur Einträge des globalen Kontexts behalten, deren Schlüssel dem Regex-Filterausdruck in Parameter `Filter` entsprechen (alle anderen Einträge löschen) • *`ListAllMatchingKeys`*: Alle Schlüssel des globalen Kontexts, die dem Regex-Filterausdruck in Parameter `Filter` entsprechen, in einem XML-Dokument ausgeben • *`ListAllNotMatchingKeys`*: Nur Schlüssel des globalen Kontexts, die nicht dem Regex-Filterausdruck in Parameter `Filter` entsprechen, in einem XML-Dokument ausgeben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.globalcontext.GlobalContextAdapter`*: Hauptklasse (Standard)
Key	Schlüssel, unter dem ein Eintrag im globalen Kontext gespeichert ist **Mögliche Werte:** Beliebige Zeichenkette

Filter	Regulärer Ausdruck (Regex), um nach Schlüsseln im globalen Kontext zu filtern, gemäß *java.util.Pattern*, siehe http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html **Mögliche Werte:** Beliebiger Regulärer Ausdruck (z.B. *.*Test.**)

Statuswerte

1	Die Operation wurde erfolgreich ausgeführt.
0	Die Operation wurde erfolgreich ausgeführt, das Ergebnis ist jedoch Null oder eine leere Liste.
-1	Der Adapter konnte nicht ausgeführt werden.

10.22.9 Keycloak Token

`Keycloak Token`: Gibt einen Keycloak-Token für einen Benutzer zurück, indem der Benutzer sich mit Benutzer und Passwort bei Keycloak an- und abmeldet.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Get Token` - Token anfordern

Parameter

`Adapter`	Hauptklasse des Adapters (Nicht verändern!) **Mögliche Werte:** *de.softproject.x4.auth.keycloak.KeycloakTokenAdapter*: Hauptklasse (Standard)
`userName`	Name des Benutzers (obligatorisch)
`password`	Passwort des Benutzers (obligatorisch)
`realm`	Bereich, der in Keycloak REST API-Aufrufen verwendet wird (optional).
`clientId`	Client ID für die Keycloak REST API-Authentifizierung (optional).
`clientSecret`	Geheimer Client-Schlüssel für die Keycloak REST API-Authentifizierung (optional).

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters ist ein Fehler aufgetreten.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

10.22.10 Log4J Logging

`Log4J Logging`: Gibt Daten aus dem Prozess über *Log4J* in das Server-Log aus. Als Input-Daten können alle Daten dienen, die sich in eine Zeichenkette umwandeln lassen. Im `AdvancedMode` können in einem XML-Dokument mit adapterspezifischer Struktur Daten auf verschiedenen Log-Stufen direkt ausgegeben werden.
Dieser Adapter gibt seine Input-Daten unverändert an den nächsten Prozess-Schritt weiter.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *`Execute`*: Adapter ausführen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.log4j.Log4jAdapter`*: Hauptklasse (Standard)
logger	Verwendeter Logger, an den die Log-Nachricht gesendet werden soll **Mögliche Werte:** • Beliebiger Logger-Name • *`X4Logger`*: Ausgabe als *`X4Logger`* in der Server-Konsole über Log4j

level	Stufe der ausgegebenen Log-Nachricht. In der Log4j-Konfiguration ist eine Schwelle (Parameter `Threshold`) definiert, ab welcher Stufe Log-Nachrichten ausgegeben werden; meist ist dies *`INFO`*. **Mögliche Werte:** • *`DEBUG`*: Debug-Informationen ausgeben (falls dies die Logging-Konfiguration zulässt) • *`INFO`*: Allgemeine Informationen oder verarbeitete Prozessdaten ausgeben • *`WARN`*: Warnung für eine unerwartete Situation ausgeben • *`ERROR`*: Fehlermeldung (abgefangene Exception) ausgeben • *`FATAL`*: Kritischen Systemfehler ausgeben ⓘ Log-Nachrichten mit dem Level `FATAL` werden technisch bedingt mit dem Level `ERROR` ausgegeben. Die Ausgabe erhält `FATAL:` als Präfix.
advancedMode	Erweiterten Modus aktivieren, in dem ein Input-XML-Dokument mit einer adapterspezifischen Struktur gesondert verarbeitet wird. Ein solches XML-Dokument kann Log-Nachrichten enthalten, die auf einer angegebenen Log-Stufe ausgegeben werden (siehe Beispiel unten) **Mögliche Werte:** • *`ja`*: Input-XML-Dokumente parsen • *`nein`*: Input-Daten nicht parsen (Standard)
modifyMessage	Log-Nachricht verändern/erweitern **Mögliche Werte:** • *`Nothing`*: Log-Nachricht mit unverändertem Layout ausgeben • *`AddPid`*: In der Log-Nachricht zusätzlich die aktuelle Prozess-Instanz-ID ausgeben (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Beispiel

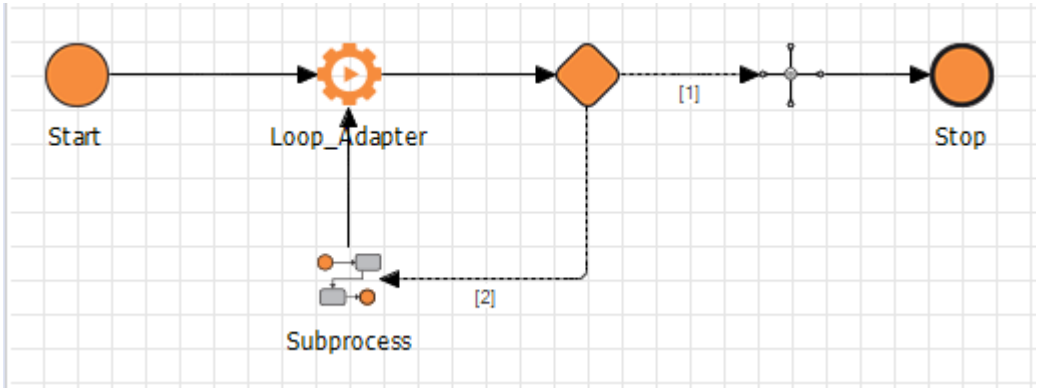
Wenn der Parameter `advancedMode` aktiviert ist: Dieses adapterspezifische Input-XML-Dokument enthält Angaben zur Log-Stufe und die auszugebenden Log-Nachrichten. Die Nachrichten werden direkt ins Server-Log ausgegeben:

```
<logs>
   <log logger="X4Logger">
      <message level="DEBUG">This is a debug message.</message>
      <message level="INFO">This is a info message. </message>
      <message level="WARN">This is a warn message. </message>
      <message level="ERROR">This is an error message. </message>
      <message level="FATAL">This is a fatal message. </message>
   </log>
</logs>
```

10.22.11 Loop Adapter

`Loop Adapter`: Der **Loop Adapter** durchläuft eine festgelegte Anzahl von Schleifendurchläufen in einem Prozess.

Sobald die festgelegte Anzahl von Schleifendurchläufen durchlaufen wurde, gibt der **Loop Adapter** den Statuswert `1` aus.



Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Count`: Zählt die Anzahl der Schleifendurchläufe.

Parameter

`count`	Anzahl der auszuführenden Schleifendurchläufe. **Mögliche Werte:** • Ganzzahl

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.22.12 Parallel Process Starter

`Parallel Process Starter`: Führt einen *X4*-Prozess in einem beliebigen Benutzerordner in `<X4>/X4DB` beliebig oft parallel aus und ermöglicht es, die Ergebnisse anschließend wieder zu synchronisieren. Der Adapter erwartet ein Input-XML-Dokument mit zu verarbeitenden Daten, jeweils abgelegt innerhalb von beliebig benannten Kindelementen unterhalb des Wurzel-Elements. Der Adapter gibt je nach Konfiguration ein XML-Dokument oder eine Base64-kodierte Zeichenkette aus.

> Mit dem *Parallel Process Starter* können Sie über *dynamische Parameter* den Prozessnamen (und -pfad) zur Laufzeit setzen. Außerdem besitzt jeder über den *Parallel Process Starter* gestartete *X4*-Prozess eine eigene Prozessinstanz-ID, wird in einem separaten Thread ausgeführt und er kann daher nicht auf Variablen des aufrufenden Prozesses zugreifen.

Allgemeine Eigenschaften und Parameter

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`Execute`*: Prozess(e) ausführen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.parallelprocessstarter.Adapter*: Hauptklasse (Standard)
X4ProcessURL	Pfad zum auszuführenden X4-Prozess **Mögliche Werte:** Beliebige gültige `x4db:`-URL zum X4-Prozess im X4 Repository (z. B. *`x4db:/1/Project/Folder/Process.wrf`*)
MaxThread	Anzahl der Threads, die genutzt werden sollen **Mögliche Werte:** • Beliebige positive ganze Zahl (z.B. *3*) • *0* oder negative ganze Zahl: parallele Ausführung nicht beschränken (z.B. *-1*)

InputIsBase64	Gibt das Datenformat für den Adapter-Input an **Mögliche Werte:** • `True`: Der Inhalt jedes Elements ist eine Base64-kodierte Zeichenkette, die dekodiert als Input an den Kind-Prozess weitergegeben wird. • `False`: Der XML-Elementinhalt wird direkt an den Kind-Prozess weitergegeben.
expandOutput	Steuert das Ausgabeformat des Adapters **Mögliche Werte:** • `True`: Das Ergebnis wird komplett als XML-Dokument ausgegeben. • `False`: Das Ergebnis wird als Base64-kodierte Zeichenkette ausgegeben.

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Im Input-XML-Dokument kann ein beliebig benanntes Wurzel-Element beliebig viele Kind-Elemente enthalten. Für jedes Kind-Element wird ein X4-Prozess aufgerufen und der Elementknoten als Input übergeben.

Beispiel-Input

```
<SomeRootElement>
  <SomeInputElement/>
  <SomeInputElement>with some content</SomeInputElement>
  <SomeInputElement someAttribute="someValue">more content</SomeInputElement>
</SomeRootElement>
```

Mit dem Parameter *`InputIsBase64:False`* wird das oben genannte Beispiel wie folgt ausgeführt:

Prozessaufruf 1

```
<SomeInputElement/>
```

Prozessaufruf 2

```
<SomeInputElement>with some content</SomeInputElement>
```

Prozessaufruf 3

```
<SomeInputElement someAttribute="someValue">more content</SomeInputElement>
```

Ausgabe

Das Ergebnis der Prozess-Ausführung könnte im obigen Beispiel wie folgt aufgebaut sein:

Output

```
 <Results>
    <Result childProcessStatus="1" childProcessPid="4711" encoding="UTF-8" content-
type="text/plain">...some base64 string...</Result>
    <Result childProcessStatus="1" childProcessPid="4712" encoding="UTF-8" content-
type="text/plain">...some base64 string...</Result>
    <Result someAttribute="someValue" childProcessStatus="1" childProcessPid="4713"
encoding="UTF-8" content-type="text/plain">...some base64 string...</Result>
</Results>
```

Im Falle eines Fehlers wird kein `Result`-Element erzeugt, sondern z.B. folgendes Ergebnis ausgegeben:

Output im Fehlerfall

```
<Error attributesFromInput="">
    <TopException class="java.lang.NullPointerException">exception message</
TopException>
    <CausedByException class="some.other.Exception">exception message</
CausedByException>
</Error>
```

10.22.13 Process Starter

`Process Starter`: Führt einen Technical Process in einem beliebigen Benutzerordner in `<X4>/X4DB` mit einer Prozessinstanz-ID synchron oder asynchron aus. Als Input erhält der gestartete Prozess den Input des Adapters. Wenn der Funktions-Adapter *asynchron* ausgeführt wird, dann gibt der Adapter unmittelbar ein XML-Dokument aus, das die Prozessinstanz-ID des gestarteten Prozesses enthält; Wenn der Funktions-Adapter *synchron* ausgeführt wird, dann wartet er auf das Ergebnis des gestarteten Technical Processes (falls vorhanden) und gibt dieses aus.

Mit dem *Process Starter* können Sie über *dynamische Parameter* den Prozessnamen (und -pfad) zur Laufzeit setzen. Außerdem besitzt jeder über den *Process Starter Adapter* gestartete Technical Process eine eigene Prozessinstanz-ID, wird in einem separaten Thread ausgeführt und er kann daher nicht auf Variablen des aufrufenden Technical Processes zugreifen.

Allgemeine Eigenschaften und Parameter

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *Run*: Prozess starten

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.processstarter.ProcessStarter*: Hauptklasse (Standard)
User	Benutzer-ID (X4DB-Benutzerordner, in dem sich der zu startende Prozess befindet) **Mögliche Werte:** • Beliebige gültige Benutzer-ID (z.B. *999*) • *1*: Standard-Benutzerordner (Standard)
Process	Repository-Pfad zu einem Technical Process, der die Eigenschaft `In service: true` besitzt **Mögliche Werte:** Beliebiger Pfad im *X4* Repository ohne führenden Schrägstrich (z.B. *Project/Folder/Process.wrf*)
synchron	Technical Process synchron oder asynchron ausführen **Mögliche Werte:** • *True*: Warten, bis die gestartete Prozessinstanz beendet ist, und ggf. deren Ergebnis ausgeben, siehe unten (Standard) • *False*: Die Prozessinstanz-ID in einem XML-Dokument ausgeben und unmittelbar den darauf folgenden Prozess-Schritt ausführen, siehe unten

forceNewThread	Auch bei synchroner Ausführung einen neuen Thread starten **Mögliche Werte:** • *True*: Bei synchroner Ausführung einen neuen Thread starten; Diese Option ermöglicht, den mit den Threads assoziierten Zustand besser zu verwalten, z.B. bei Transaktionen. (Standard) • *False*: Bei synchroner Ausführung keinen neuen Thread starten (ursprüngliches Verhalten)

Synchron gestartete Technical Processes

Wenn Sie Technical Processes synchron starten (mit Parameter `synchron: True`), dann wartet der Adapter, bis der gestartete Technical Process vollständig ausgeführt wurde und gibt ggf. dessen Ergebnis inklusive Statuscode aus, bevor der nächste Prozess-Schritt ausgeführt wird.

Statuswerte

`1`	(Oder jeder andere positive Wert) Der Adapter wurde erfolgreich *synchron* ausgeführt. Der Statuscode des synchron gestarteten Technical Processes wird vom *Process Starter* übernommen oder ggf. verändert ausgegeben (siehe Hinweis unten). Wenn der Adapter Status *1* ausgibt, heißt das nicht notwendigerweise, dass der gestartete Technical Process ein Ergebnis ausgibt.
`0`	Der zu startende Technical Process ist entweder nicht aktiv (Eigenschaft `In service: false`) und wurde daher nicht ausgeführt, oder im gestarteten Technical Process werden leere Bausteine verwendet, oder der gestartete Technical Process gibt den Status *0* aus.
`-1`	Der Funktions-Adapter konnte nicht ausgeführt werden (der zu startende Technical Process wurde nicht gefunden oder der gestartete Technical Process wirft Fehler).
(Jeder negative Wert)	Der gestartete Technical Process wurde möglicherweise ausgeführt und erzeugte einen negativen Statuswert, z.B. über ein XSL-Mapping mit `ConditionState`, siehe Dynamische Statuswerte mit `ConditionState`. Bei asynchron gestarteten Technical Processes wird deren Status vom *Process Starter* übernommen. Die weitere Prozessausführung wird abgebrochen, falls die Technical Process-Eigenschaft `Stop on error: true` ist.

Beachten Sie:

- Wenn ein synchron gestarteter Technical Process einen Statuswert zwischen *950* und *999* zurückgibt, wird dieser Status vom *Process Starter* verändert (Statuswert minus 50) und es werden Statuswerte zwischen *900* und *949* ausgegeben. Dies ist erforderlich, um die *Save Point*-Funktion des *X4 Servers* nicht zu beeinflussen.
- Wenn ein synchron gestarteter Technical Process einen Statuswert zwischen *-999* und *-950* zurückgibt, wird dieser Status vom *Process Starter* verändert (Statuswert plus 50) und es werden Statuswerte zwischen *-900* und *-949* ausgegeben. Dies ist erforderlich, um Fehler in der Prozess-Ausführung zu vermeiden.

Asynchron gestartete Technical Process

Wenn Sie Technical Process asynchron starten (Parameter `synchron`: *False*), dann wird nach dem Start des aufgerufenen Technical Process dessen Prozessinstanz-ID in einem XML-Dokument ausgeben und der unmittelbar darauf folgende Prozess-Schritt ausgeführt.

Statuswerte

1 (successful)	Der Funktions-Adapter wurde erfolgreich *asynchron* ausgeführt. Wenn der gestartete Technical Process nicht aktiv ist (Eigenschaft In service: *false*) oder Fehler wirft, dann wird dennoch Status *1* ausgegeben; Das Ergebnis-XML-Dokument enthält in diesen Fällen die Prozessinstanz-ID *-1*.
-1 (error)	Der Funktions-Adapter konnte nicht ausgeführt werden (der zu startende Technical Process wurde nicht gefunden).

10.22.14 Projects

`Projects`: Ermöglicht den Zugriff auf Projekte innerhalb des X4 Repositorys.

Dieser Adapter löst den Repository Explorer (veraltet) ab.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Create`: Erzeugt ein neues Projekt. **Erforderliche Parameter:** `projectName`, `projectType` • `List`: Erzeugt eine Liste aller vorhandenen Projekte. • `Delete`: Löscht ein Projekt. **Erforderliche Parameter:** `projectName`

Parameter

`Adapter`	Hauptklasse des Adapters (Nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.system.repository.ProjectAdapter*: Hauptklasse (Standard)
`projectName`	Name des Projekts, das erstellt oder gelöscht werden soll. **Mögliche Werte:** Zeichenkette ohne Leerzeichen und Sonderzeichen.
`projectType`	Art des Projektes, das erstellt werden soll. **Mögliche Werte:** • *`ESB Project`* • *`Web App Project`* • *`Adapter Project`* • *`Web Portal Project`*

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters ist ein Fehler aufgetreten.

Input

Der Adapter erwartet keinen Input. Vorhandener Input wird unverändert ausgegeben.

Output

- Der Adapter liefert bei den Operationen `Create` und `Delete` nur im Fehlerfall einen Output.
- Operation `List`:

Beispiel-Output für die Operation List

```
<Projects>
    <Project type="ESB" name="MyESBProject" />
    <Project type="BPM" name="MyBPMProject" />
</Projects>
```

10.22.15 Project Access

`Project Access`: Ermöglicht den Zugriff auf Dateien innerhalb eines Projektes, in Prozessbibliotheken und Adaptern.

> Dieser Adapter löst den Repository Explorer (veraltet) ab.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `ReadFile`: Im Parameter `path` festgelegte Datei lesen. • `WriteFile`: Daten aus dem vorangegangenen Prozess-Schritt in die im Parameter `path` angegebene Zieldatei schreiben. • `AppendFile`: Datei aus dem vorangegangenen Prozess-Schritt in die im Parameter `path` angegebene Zieldatei anhängen. > Wenn die im Parameter `path` festgelegte Zieldatei nicht existiert, wird eine Datei erstellt. • `Exists`: Prüfen, ob die im Parameter `path` festgelegten Dateien oder Ordner existieren. Gibt als Ergebnis eine XML mit Metadaten aus. • `CreateDirectory`: Im Parameter `path` festgelegten Ordner erstellen. • `Delete`: Im Parameter `path` festgelegte Dateien oder Ordner löschen. • `ListDirectory`: Namen und Metadaten von Dateien und Ordnern aus dem im Parameter `path` festgelegten Ordner auflisten. Gibt als Ergebnis eine XML mit Metadaten aus. > Der Inhalt der Dateien wird nicht gelistet. • `Move`: Im Parameter `path` festgelegte Dateien oder Ordner in den im Parameter `targetPath` festgelegten Ordner verschieben. • `Copy`: Im Parameter `path` festgelegte Dateien oder Ordner in den im Parameter `targetPath` festgelegten Ordner kopieren.

Parameter

`Adapter`	Hauptklasse des Adapters (Nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.system.repository.ProjectAccessAdapter*: Hauptklasse (Standard)
`path`	Pfad zur Datei bzw. dem Ordner, der verarbeitet werden soll. **Mögliche Werte:** Verzeichnispfad ⓘ Der Projektpfad muss ohne den Projektnamen angegeben werden. **Beispiel**: *Processes/MyFolder*
`targetPath`	Für Operationen `copy` und `move`: Ziel der zu kopierenden oder zu verschiebenden Datei/Ordner
`nameFilter`	Filtert den Output der Operation `ListDirectory` nach Name **Mögliche Werte:** • Beliebige Zeichenkette (z.B. *`myFolder`*) • Platzhalter (Wildcard) `*`

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`0`	Für die Operation `Exists` wurde keine Datei gefunden.
`-1`	Bei der Ausführung des Adapters ist ein Fehler aufgetreten.

Weitere Informationen finden Sie im Abschnitt

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.1 Operation: ReadFile

Die Operation `ReadFile` liest die Daten aus der im Parameter `path` festgelegten Datei.

10.22.15.1.1 Beispiele

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `ReadFile`, um die Daten aus der im Parameter `path` angegebenen Zieldatei zu lesen.

Parameter	Wert
`path`	Resources/File.xml

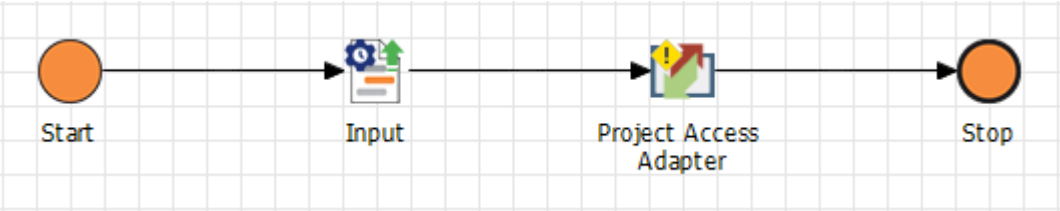
Weitere Informationen finden Sie im Abschnitt

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.2 Operation: WriteFile

Die Operation `WriteFile` schreibt Daten aus dem vorangegangenen Prozess-Schritt in die in Parameter `path` angegebene Zieldatei im X4 Repository.

10.22.15.2.1 Beispiele



Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `WriteFile`, um die Daten aus dem vorangegangenen Prozess-Schritt in die in Parameter `path` angegebene Zieldatei zu schreiben.

Parameter	Wert
`path`	Resources/File.xml

Weitere Informationen finden Sie im Abschnitt

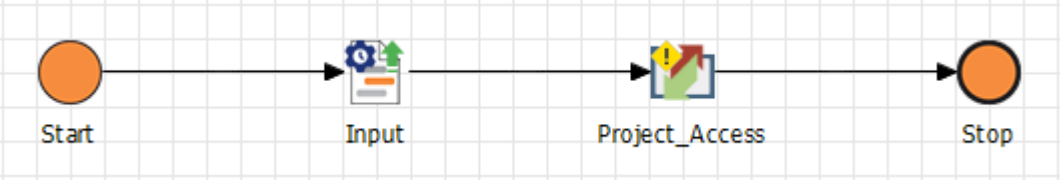
- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.3 Operation: AppendFile

Die Operation `AppendFile` hängt die Daten aus dem vorangegangenen Prozess-Schritt in die in Parameter `path` angegebene Zieldatei im X4 Repository.

Wenn die im Parameter `path` festgelegte Zieldatei nicht existiert, wird eine Datei erstellt.

10.22.15.3.1 Beispiele



Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `AppendFile`, um die Daten aus dem vorangegangenen Prozess-Schritt in die in Parameter `path` angegebene Zieldatei anzuhängen.

Parameter	Wert
`path`	Resources/File.xml

10.22.15.3.1.1 Zieldatei

```
<?xml version="1.0" encoding="UTF-8"?>
<Cars>
    <Car>
        <Brand>Audi</Brand>
        <Country>Germany</Country>
    </Car>
</Cars>
```

10.22.15.3.1.2 Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Cars>
    <Car>
        <Brand>BMW</Brand>
        <Country>Germany</Country>
    </Car>
    <Car>
        <Brand>Hyundai</Brand>
        <Country>South Korea</Country>
    </Car>
</Cars>
```

10.22.15.3.1.3 Output

```
<?xml version="1.0" encoding="UTF-8"?>
<Cars>
    <Car>
        <Brand>Audi</Brand>
        <Country>Germany</Country>
    </Car>
</Cars><?xml version="1.0" encoding="UTF-8"?>
<Cars>
    <Car>
        <Brand>BMW</Brand>
        <Country>Germany</Country>
    </Car>
    <Car>
        <Brand>Hyundai</Brand>
        <Country>South Korea</Country>
    </Car>
</Cars>
```

⚠ Bei dem Output handelt es sich um eine nicht valide XML-Datei. Der Output muss im Anschluss mithilfe eines Mappings in eine valide XML-Datei transformiert werden.

Weitere Informationen finden Sie im Abschnitt

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.4 Operation: Exists

Die Operation `Exists` prüft, ob die im Parameter `path` festgelegten Dateien oder Ordner existieren. Gibt als Ergebnis eine XML mit Metadaten aus.

Statuswerte

`1`	Die Datei oder der Ordner existiert.
`0`	Die Datei oder der Ordner existiert nicht.

10.22.15.4.1 Beispiele

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Exist`, um zu prüfen, ob die in Parameter `path` angegebene Zieldatei existiert.

Parameter	Wert
`path`	Resources/File.xml

Weitere Informationen finden Sie im Abschnitt

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.5 Operation: CreateDirectory

Die Operation `CreateDirectory` erstellt den im Parameter `path` festgelegten Ordner.

10.22.15.5.1 Beispiele

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `CreateDirectory`, um den im Parameter `path` angegebenen Order zu erstellen.

Parameter	Wert
`path`	TemporaryFiles/MyFolder

Weitere Informationen finden Sie im Abschnitt

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.6 Operation: Delete

Die Operation `Delete` löscht die im Parameter `path` festgelegte lokale Datei oder Ordner.

Die Daten werden unwiderruflich gelöscht.

10.22.15.6.1 Beispiele

10.22.15.6.1.1 Datei löschen

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Delete`, um die im Parameter `path` angegebene Zieldatei zu löschen.

Parameter	Wert
`path`	Resources/File.xml

10.22.15.6.1.2 Ordner löschen

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Delete`, um den im Parameter `path` angegebenen Zielordner zu löschen.

Parameter	Wert
`path`	Resources/MyFolder

Weitere Informationen finden Sie im Abschnitt

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.7 Operation: ListDirectory

Die Operation `ListDirectory` listet Namen und Metadaten von Dateien und Ordnern aus dem im Parameter `path` festgelegten Ordner auf. Gibt als Ergebnis eine XML mit Metadaten aus.

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Der Ordner existiert nicht.

10.22.15.7.1 Beispiele

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `ListDirectory`, um PDF-Dateien aus dem im Parameter `path` festgelegten Ordner aufzulisten.

Parameter	Wert
`path`	Resources/MyFolder
`nameFilter`	*.pdf

10.22.15.7.1.1 Output

```
<?xml version="1.0" encoding="UTF-8"?>
<Entries>
   <Entry isDirectory="false" name="Document v1.pdf">
      <CreationDate>2021-06-18T09:40:39.1965Z</CreationDate>
      <LastModified>2021-06-18T09:20:49.137675Z</LastModified>
   </Entry>
   <Entry isDirectory="false" name="Document v2.pdf">
      <CreationDate>2021-06-18T09:40:07.105917Z</CreationDate>
      <LastModified>2021-06-18T09:20:49.137675Z</LastModified>
   </Entry>
   <Entry isDirectory="false" name="Document v3.pdf">
      <CreationDate>2021-06-18T09:40:18.941018Z</CreationDate>
      <LastModified>2021-06-18T09:20:49.137675Z</LastModified>
   </Entry>
</Entries>
```

Wenn ein Ordner gelistet wird, wird das Attribut `isDirectory="true"` gesetzt.

Weitere Informationen finden Sie im Abschnitt

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.8 Operation: Move

Die Operation `Move` verschiebt die im Parameter `path` festgelegte Datei oder Ordner in den im Parameter `targetPath` festgelegten Ordner.

10.22.15.8.1 Beispiele

10.22.15.8.1.1 Ordner verschieben

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Move`, um den Ordner `MyFolder` in das Verzeichnis `X4` zu verschieben:

Parameter	Wert
`path`	Resources/MyFolder
`targetPath`	Resources/X4/MyFolder

10.22.15.8.1.2 Datei verschieben

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Move`, um die Datei `MyFile.xml` in das Verzeichnis `MyFolder` zu verschieben:

Parameter	Wert
`path`	Resources/MyFile.xml
`targetPath`	Resources/MyFolder/MyFile.xml

Weitere Informationen finden Sie im Abschnitt

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.9 Operation: Copy

Die Operation `Copy` kopiert die im Parameter `path` festgelegte Datei oder Ordner in den im Parameter `targetPath` festgelegten lokalen Ordner.

10.22.15.9.1 Beispiele

10.22.15.9.1.1 Ordner kopieren

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Copy`, um den Ordner `MyFolder` in das Verzeichnis `X4` zu kopieren:

Parameter	Wert
`path`	Resources/MyFolder
`targetPath`	Resources/X4/MyFolder

10.22.15.9.1.2 Datei kopieren

Im folgenden Beispiel sehen Sie eine beispielhafte Parametrisierung des Adapters für die Operation `Copy`, um die Datei `MyFile.xml` in das Verzeichnis `MyFolder` zu kopieren:

Parameter	Wert
`path`	Resources/MyFile.xml
`targetPath`	Resources/MyFolder/MyFile.xml

Weitere Informationen finden Sie im Abschnitt

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.16 System Status

`System Status`: Der **System Status** Adapter gibt Statusinformationen zum X4 Server aus.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Threads Information`: Gibt Informationen über alle auf dem X4 Server laufenden Threads aus • `JVM Parameters`: Gibt alle Parameter der Java Virtual Machine aus • `Processes Running`: Gibt Informationen über alle auf dem X4 Server laufenden Prozesse aus • `User Sessions Memory`: Gibt Informationen zum Speicherverbrauch der Benutzersitzungen aus • `Apps Status`: Gibt Informationen zu den fehlgeschlagenen und bereitgestellten X4 Apps aus

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.

`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Weitere Informationen finden Sie im Abschnitt

- Operation: Threads Information
- Operation: JVM Parameters
- Operation: Processes Running
- Operation: User Sessions Memory
- Operation: Apps Status

10.22.16.1 Operation: Threads Information

Die Operation `Threads Information` gibt Informationen über alle auf dem X4 Server laufenden Threads aus.

10.22.16.1.1 Beispiele

10.22.16.1.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ThreadsInformation>
    <Thread>
        <Name>MSC service thread 1-3</Name>
        <Id>18</Id>
        <State>WAITING</State>
        <Priority>5</Priority>
        <Daemon>false</Daemon>
        <CpuTime>17</CpuTime>
    </Thread>
    <Thread>
        <Name>X4 License Recheck Timer</Name>
        <Id>224</Id>
        <State>TIMED_WAITING</State>
        <Priority>5</Priority>
        <Daemon>true</Daemon>
        <CpuTime>4</CpuTime>
    </Thread>
</ThreadsInformation>
```

Folgende Elemente sind im Output enthalten:

Element	Beschreibung	Mögliche Werte
`Name`	Name des Threads	• Beliebiger String

Element	Beschreibung	Mögliche Werte
`Id`	Id des Threads	• Ganzzahl
`State`	Status des Threads	• `NEW` • `RUNNABLE` • `BLOCKED` • `WAITING` • `TIMED_WAITING` • `TERMINATED`
`Priority`	Priorität des Threads	• Ganzzahl zwischen `1` und `10`
`Daemon`	Angabe, ob der Thread ein Hintergrunddienst ist	• `true` • `false`
`CpuTime`	Für die Ausführung des aktuellen Threads benötigte Zeit in Sekunden	• Ganzzahl

Weitere Informationen finden Sie im Abschnitt

- Operation: Threads Information
- Operation: JVM Parameters
- Operation: Processes Running
- Operation: User Sessions Memory
- Operation: Apps Status

10.22.16.2 Operation: JVM Parameters

Die Operation `JVM Parameters` gibt alle Parameter der Java Virtual Machine aus.

10.22.16.2.1 Beispiel

10.22.16.2.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<JVMParameters Unit="MB">
    <MaxMemory>2048</MaxMemory>
    <CurrentMemory>1940</CurrentMemory>
    <FreeMemory>720</FreeMemory>
    <AvailableProcessors>16</AvailableProcessors>
</JVMParameters>
```

Folgende Elemente sind im Output enthalten:

Element	Beschreibung	Mögliche Werte
`MaxMemory`	Maximal verfügbarer Arbeitsspeicher (in Megabyte)	beliebige Ganzzahl
`CurrentMemory`	Lokaler Arbeitspeicher, einschließlich genutztem und ungenutztem Arbeitsspeicher (in Megabyte)	beliebige Ganzzahl
`FreeMemory`	Ungenutzter Arbeitsspeicher (in Megabyte)	beliebige Ganzzahl
`AvailableProcessors`	Anzahl der verfügbaren Prozessoren	beliebige Ganzzahl

Weitere Informationen:

- Operation: Threads Information
- Operation: JVM Parameters
- Operation: Processes Running
- Operation: User Sessions Memory
- Operation: Apps Status

10.22.16.3 Operation: Processes Running

Die Operation `Processes Running` gibt Informationen über alle auf dem X4 Server laufenden Prozesse aus.

10.22.16.3.1 Beispiel

10.22.16.3.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ProcessesRunning>
    <Process>
        <UUID>9e18ecdf-92c4-4eda-ae48-e55c49bdcca5</UUID>
        <Name>MyProcess.wrf</Name>
        <Path>1/SystemStatus/Processes/MyProcess.wrf</Path>
        <Module>engine</Module>
    </Process>
    <Process>
        <UUID>c6164963-a092-408f-8abf-f157ca87cd42</UUID>
        <Name>MySubProcess.wrf</Name>
        <Path>1/SystemStatus/Processes/MyFolder/MySubProcess.wrf</Path>
        <Module>engine</Module>
    </Process>
    <MemoryConsumed Unit="B">254</MemoryConsumed>
</ProcessesRunning>
```

Folgende Elemente sind im Output enthalten:

Element	Beschreibung	Mögliche Werte
`UUID`	UUID des Prozesses	UUID
`Name`	Name des Prozesses	Beliebiger String
`Path`	Pfad des Prozesses	Beliebiger String
`Module`	Name des Moduls, das den Prozess ausführt	`engine`
`MemoryConsumed`	Alle von den Prozessen belegte Arbeitsspeicher in Byte	`Double`

Weitere Informationen:

- Operation: Threads Information
- Operation: JVM Parameters
- Operation: Processes Running
- Operation: User Sessions Memory
- Operation: Apps Status

10.22.16.4 Operation: User Sessions Memory

Die Operation `User Sessions Memory` gibt Informationen zum Speicherverbrauch der Benutzersitzungen aus.

10.22.16.4.1 Beispiel

10.22.16.4.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<UserSessionMemory Unit="B">
    <UserSession>
        <SessionId>f6e9a71a-a583-4642-af90-5e68e422f5a9</SessionId>
        <Size>0</Size>
    </UserSession>
</UserSessionMemory>
```

Folgende Elemente sind im Output enthalten:

Element	Beschreibung	Mögliche Werte
`SessionId`	ID der Benutzersitzung	UUID

Element	Beschreibung	Mögliche Werte
`Size`	Von der Benutzersitzung belegter Arbeitsspeicher in Byte	Integer

Weitere Informationen:

- Operation: Threads Information
- Operation: JVM Parameters
- Operation: Processes Running
- Operation: User Sessions Memory
- Operation: Apps Status

10.22.16.5 Operation: Apps Status

Die Operation `Apps Status` gibt Informationen zu den fehlgeschlagenen und bereitgestellten X4 Apps aus.

10.22.16.5.1 Beispiele

10.22.16.5.1.1 Output

Der Adapter erzeugt ein XML-Dokument als Output:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AppStatus>
    <AppsDeployed>
        <Number>1</Number>
        <App>MyX4App.x4app</App>
    </AppsDeployed>
    <AppsFailed>
        <Number>1</Number>
        <App>MyX4App.x4app</App>
    </AppsFailed>
</AppStatus>
```

Folgende Elemente sind im Output enthalten:

Element	Beschreibung	Mögliche Werte
Number	Anzahl der bereitgestellten X4 Apps	• Ganzzahl
App	Name der X4 App-Datei inklusive der Dateiendung	• Beliebiger String

Weitere Informationen finden Sie im Abschnitt

- Operation: Threads Information
- Operation: JVM Parameters
- Operation: Processes Running
- Operation: User Sessions Memory
- Operation: Apps Status

10.22.17 Task Management

`Task Management`: Ermöglicht den schnittstellenbasierten Zugriff auf Human Tasks in Business Processes mittels Technical Processes. Er bildet eine Schnittstelle, die modellierte Aufgaben und menschliche Interaktion auf Prozessebene verfügbar macht. Dies ermöglicht die Anbindung von kundenspezifischen Web Apps an Vorgänge, die mit BPMN modelliert werden oder auch die direkte Anbindung beliebiger Drittsysteme an die zentrale Vorgangssteuerung der X4 BPMS.

Für den Zugriff mit dem Adapter wird der angemeldete Benutzer in X4 Web Apps verwendet.

Eigenschaften

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • `GetAllTasks`: Alle Tasks für den angemeldeten Benutzer oder die Gruppe zurückgeben • `GetTask`: Einen Task für den angemeldeten Benutzer zurückgeben • `EditTask`: Die Aufgaben für andere Benutzer sperren und die Merkmale aktualisieren • `CloseTask`: Die Aufgabe mit einem Fachstatus abschließen und den Vorgang fortsetzen • `DeclineTask`: Die Aufgabe ablehnen und an den Eigentümer zurückgeben • `ForwardTask`: Die Aufgabe an eine Person oder Gruppe weitergeben • `CancelTask`: Die Bearbeitung der Aufgabe abbrechen und sie für andere Benutzer freigeben • `ResubmitTask`: Aufgabe auf Wiedervorlage setzen

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.x4.server.bpm.casemanagement.adapter.TaskManagement*: Hauptklasse (Standard)

`limit`	Seitengröße für den Abruf der Task-Liste Der Wert ist optional. Ist der Parameter nicht gesetzt, werden alle Tasks ausgegebenen. **Mögliche Werte:** Beliebige Zahl *> 0*
`offset`	Offset für den Abruf der Task-Liste Der Wert ist optional. Ist der Parameter nicht gesetzt, werden alle Tasks ausgegebenen. **Mögliche Werte:** Beliebige Zahl *> 0*
`state`	*Erforderlich*. Zustandsfilter für den Abruf der Task-Liste **Mögliche Werte:** • *OPEN*: Offene Tasks • *IN_PROGRESS*: Task in Bearbeitung • *CLOSED*: Geschlossene Tasks
`taskUuid`	UUID des Tasks **Mögliche Werte:** Gültige UUID, z. B. *443c498a-5868-4110-891f-42e5cb8fc24e*
`userForDebugging`	Benutzer der zum Debugging verwendet werden soll, wenn keine Anmeldung in X4 Web Apps vorhanden ist **Mögliche Werte:** Gültiges Benutzer-Login, z.B. *admin*
`dataformatVersion`	Version des verwendeten Datenformats **Mögliche Werte:** • `1.0`: Verwendetes Datenformat bis Version 7.0 • `2.0`: Verwendetes Datenformat ab Version 7.0, siehe Input-Beispiele

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt
`-1`	Die Operation konnte nicht ausgeführt werden

Die Dokumentation der ReST-Schnittstelle finden Sie unter http://localhost:8080/X4/X4Api/.

Input

Bei folgenden Operationen wird kein Input benötigt:

- `GetAllTasks`
- `GetTask`
- `CancelTask`

Folgende Operationen benötigen als Input ein XML-Dokument mit folgendem Aufbau:

- Operation `EditTask`

  Beispiel-Input für die Operation EditTask

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <EditTask>
      <Offer> 150 Mio. €</Offer>
  </EditTask>
  ```

- Operation `CloseTask`

  Beispiel-Input für die Operation CloseTask

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <CloseTask>
      <MarketValue>120 Mio. €</MarketValue>
      <Action>Declined</Action>
  </CloseTask>
  ```

- Operation `DeclineTask`

  Beispiel-Input für die Operation DeclineTask

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <DeclineTask>
      <Comment>string</Comment>
  </DeclineTask>
  ```

- Operation `ForwardTask`

  Beispiel-Input für die Operation ForwardTask

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <ForwardTask>
      <Comment>string</Comment>
      <Deadline>2019-04-15T11:46:02.782Z</Deadline>
      <Assignment type="USER|GROUP">string</Assignment>
  </ForwardTask>
  ```

Ausgabe

Bei der Operation `CancelTask` wird als Output das Input-Dokument ausgegeben.

Folgende Operationen geben als Output ein XML-Dokument mit folgender Struktur aus:

- Operation `GetAllTasks`

```
<?xml version="1.0" encoding="UTF-8"?>
<Tasks>
    <Task created="2020-12-21T09:10:53Z" creator="System" id="81f4abb5-
c2f5-418e-acf5-654304234bf4">
        <Title>humantask</Title>
        <Owner>admin</Owner>
        <Assignment type="USER">admin</Assignment>
        <State>OPEN</State>
        <KeyIdentifier/>
        <CaseId>00860a8a-12b9-45b8-b271-ec5d42573c59</CaseId>
    </Task>
</Tasks>
```

- Operationen `GetTask`, `EditTask`, `CloseTask`, `DeclineTask` und `ForwardTask`

```
<?xml version="1.0" encoding="UTF-8"?>
<Task created="2020-12-21T09:10:53Z" creator="System"
      id="81f4abb5-c2f5-418e-acf5-654304234bf4">
    <Title>humantask</Title>
    <Owner>admin</Owner>
    <Assignment type="USER">admin</Assignment>
    <State>OPEN</State>
    <KeyIdentifier/>
    <CaseId>00860a8a-12b9-45b8-b271-ec5d42573c59</CaseId>
    <Actions>
        <Action displayName="done" name="Accepted"/>
        <Action displayName="Declined" name="Declined"/>
    </Actions>
    <DecliningAllowed>false</DecliningAllowed>
    <ForwardingAllowed>false</ForwardingAllowed>
    <Comments/>
    <BusinessData>
        <Company>Max Mustermann Ink.</Lastname>
        <MarketValue>100 Mio. €</MarketValue>
        <Offer>150 Mio. €</Offer>
    </BusinessData>
</Task>
```

10.22.18 User Identity Adapter

`User Identity Adapter`: Ermöglicht es, Informationen über den aktuell angemeldeten Benutzer abzurufen, z. B. Benutzername, Vor- und Nachname, eingestellte Sprache, E-Mail-Adresse und Gruppen, denen dieser Benutzer angehört.

⚠ Wenn ein vom Standard abweichender Realm verwendet wird, muss im Client ein Mapper des Typs **Group Membership** eingerichtet werden. Die Konfiguration finden Sie im Administrationshandbuch, im Abschnitt Standardkonfiguration.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `GetUserIdentity`: Gibt die Informationen über den Benutzer zurück.

Parameter

`Adapter`	Adapter Hauptklasse (nicht ändern!) **Mögliche Werte:** • *`de.softproject.x4.adapter.useridentity.UserIdentityAdapter`*: Hauptklasse (default)
`generateTestGroup`	Legt fest, ob ein Test-Gruppeneintrag generiert werden soll, wenn der Adapter im Debug-Modus ausgeführt wird. ℹ Funktioniert ausschließlich im Debug-Modus. **Mögliche Werte:** • *`true`*: Beispielhaften Gruppeneintrag generieren • *`false`*: Keinen beispielhaften Gruppeneintrag generieren
`testGroupName`	Technischer Name des Test-Gruppeneintrags, wenn `generateTestGroup` auf *`true`* gesetzt ist.
`testGroupDisplayName`	Angezeigter Name des Test-Gruppeneintrags, wenn `generateTestGroup` auf *`true`* gesetzt ist.
`username`	Verwendeter Benutzername, wenn `generateTestGroup` auf *`true`* gesetzt ist.
`lastname`	Verwendeter Nachname, wenn `generateTestGroup` auf *`true`* gesetzt ist.
`firstname`	Verwendeter Vorname, wenn `generateTestGroup` auf *`true`* gesetzt ist.
`email`	Verwendete E-Mail-Adresse, wenn `generateTestGroup` auf *`true`* gesetzt ist.

`timezone`	Benutzer-Zeitzone gemäß ISO 8601, die im Debug-Modus ausgegeben wird. **Mögliche Werte:** Zeitzone gemäß ISO 8601 duration format, z. B. *PT5H*

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters ist ein Fehler aufgetreten.

10.22.18.1 Output-XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Result>
    <Username></Username>
    <Language></Language>
    <Groups>
        <Group>
            <Name></Name>
            <DisplayName></DisplayName>
        </Group>
        <Group>
            <Name></Name>
            <DisplayName></DisplayName>
        </Group>
    </Groups>
    <Timezone>
        <Offset></Offset>
    </Timezone>
    <AccessToken></AccessToken>
    <SessionId></SessionId>
    <Claims>
        <Claim key="" value=""/>
        <Claim key="" value=""/>
        <ComplexClaim key="" type="">
            <ComplexClaim key="" type="">
                <ComplexClaim key="" type="">
                    <Entry value=""/>
                    <Entry value=""/>
                    <Entry value=""/>
                </ComplexClaim>
                <isParent></isParent>
            </ComplexClaim>
            <isParent></isParent>
        </ComplexClaim>
    </Claims>
</Result>
```

10.22.18.2 XSD

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="
http://www.w3.org/2001/XMLSchema">
    <xs:element name="Result" type="ResultType"/>
    <xs:complexType name="GroupType">
        <xs:sequence>
            <xs:element type="xs:string" name="Name"/>
            <xs:element type="xs:string" name="DisplayName"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="GroupsType">
        <xs:sequence>
            <xs:element type="GroupType" name="Group" maxOccurs="unbounded" minOccurs="
0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="TimezoneType">
        <xs:sequence>
            <xs:element type="xs:duration" name="Offset"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ClaimType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute type="xs:string" name="key" use="optional"/>
                <xs:attribute type="xs:string" name="value" use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="EntryType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute type="xs:string" name="value" use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="ComplexClaimType">
        <xs:sequence>
            <xs:element type="EntryType" name="Entry" maxOccurs="unbounded" minOccurs="
0"/>
            <xs:element type="ComplexClaimType" name="ComplexClaim" minOccurs="0"/>
            <xs:element type="ClaimType" name="Claim" maxOccurs="unbounded" minOccurs="
0"/>
        </xs:sequence>
        <xs:attribute type="xs:string" name="key" use="optional"/>
        <xs:attribute type="xs:string" name="type" use="optional"/>
    </xs:complexType>
    <xs:complexType name="ClaimsType">
        <xs:sequence>
            <xs:element type="ClaimType" name="Claim" maxOccurs="unbounded" minOccurs="
0"/>
```

```
            <xs:element type="ComplexClaimType" name="ComplexClaim" maxOccurs="unboun
ded" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ResultType">
        <xs:sequence>
            <xs:element type="xs:string" name="Username"/>
            <xs:element type="xs:string" name="Language"/>
            <xs:element type="GroupsType" name="Groups"/>
            <xs:element type="TimezoneType" name="Timezone"/>
            <xs:element type="xs:string" name="AccessToken"/>
            <xs:element type="xs:string" name="SessionId"/>
            <xs:element type="ClaimsType" name="Claims"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

10.22.19 User Session Adapter

`User Session Adapter`: Ermöglicht es, die Daten im Datenstrom in die User Session zu schreiben, Daten aus der User Session zu lesen und aufzulisten und die User Session zu invalidieren.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `Store Content`: Schreibt die Daten im Datenstrom in die User Session. • `Get Content`: Gibt den im Parameter `key` festgelegten Datensatz aus der User Session aus. • `List Content`: Gibt eine Liste der Daten in der User Session aus. • `Check User Session`: Prüft, ob eine User Session existiert und gibt die User Session ID aus. • `Clear Session data`: Löscht die Daten in der User Session und invalidiert die User Session

Parameter

`key`	Bei Operation `Get User Session`: Wird als Filter verwendet. Bei Operation `Store User Session`: Wird als Name verwendet, mit dem die Daten im Datenstrom in der User Session gespeichert werden. **Mögliche Werte:** • Beliebiger String

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters ist ein Fehler aufgetreten.
`0`	Die Adapter-Operation hat kein Ergebnisse geliefert.

10.22.20 Web App Translation

`Web App Translation`: Gibt die lokalisierten Werte für definierte Translation Keys zurück.

Eigenschaften (Properties)

`Operation`	Vom Adapter ausgeführte Operation **Mögliche Werte:** • `GetTranslation`: Gibt die lokalisierten Werte für definierte Translation Keys zurück.

Parameter

`Adapter`	Adapter Hauptklasse (nicht ändern!) **Mögliche Werte:** • *`de.softproject.x4.server.webapp.translation.adapter.TranslationAdapter`*: Hauptklasse (Standard)
`language`	Sprache, für die die Werte ausgelesen werden. **Mögliche Werte:** Zeichenkette mit dem definierten Sprachschlüssel aus der Web-App-Definition
`keys`	Translation Key/-s, der/die ausgelesen werden sollen. Wenn mehrere Translation Keys ausgelesen werden, müssen diese durch **;** getrennt werden. **Mögliche Werte:** Zeichenkette mit der/den definierten Translation Key/-s aus der Sprachdatei.

Statuswerte

`1`	Die Adapter-Operation wurde erfolgreich ausgeführt.
`-1`	Bei der Ausführung des Adapters ist ein Fehler aufgetreten.

Ausgabe

- Sprachdatei wurde erfolgreich ausgelesen:

```
<?xml version="1.0" encoding="UTF-8"?>
<Result>
    <Output key="loginLabel" value="User"/>
    <Output key="passwordLabel" value="Password"/>
</Result>
```

- Es wurde keine Sprache gesetzt:

```
<?xml version="1.0" encoding="UTF-8"?>
<Result>No language set</Result>
```

- Es wurde kein Translation Key gesetzt:

```
<?xml version="1.0" encoding="UTF-8"?>
<Result>No keys set</Result>
```

⚠ Wenn ein Translation Key, der nicht definiert ist, abgefragt wird, wird statt des Wertes der Translation Key selbst zurückgegeben. Wenn eine Sprachdatei, die nicht existiert abgefragt werden soll, wird ebenfalls der Translation Key selbst zurückgegeben.

10.22.21 X4 Management

`X4 Management`: Ruft die Methoden des *X4*-spezifischen *JMX*-MBeans X4 Management auf. Verfügbar sind die Methoden *`SetAllOutOfService`* und *`SetAllInService`*.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`SetAllOutOfService`*: Setzt bei *allen* Prozessen die Eigenschaft `In service` auf *`false`*. Dies bewirkt, dass keine Prozesse mehr gestartet werden. • *`SetAllInService`*: Setzt bei *allen* Prozessen die Eigenschaft `In service` auf *`true`*. Dies bewirkt, dass alle Prozesse (wieder) gestartet werden können.

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.x4management.X4ManagementCheckerAdapter`*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.22.22 X4 Server Config Reader

`X4 Server Config Reader`: Gibt Java-Parameter aus dem *X4*-Startup-Skript sowie *X4*-Konfigurations-Eigenschaften aus der Datei `X4config.xml` in einem XML-Dokument aus. Die gewünschten Parameter werden dabei in einem spezifisch aufgebauten Input-XML-Dokument genannt und vom Adapter um die entsprechenden Werte ergänzt wieder ausgegeben.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`GetProperties`*: Konfigurations-Eigenschaften ausgeben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.ConfigReader`*: Hauptklasse (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Input

Dieser Funktions-Adapter erwartet eine adapterspezifische Input-XML-Struktur. Es können beliebig viele `<Param/>`-Elemente eingetragen werden; das Element `<X4Config/>` oder `<JavaConfig/>` kann je nach Bedarf weggelassen werden.

```
<Config>
   <X4Config>
       <Param><Key><!--Element in X4Config.xml--></Key></Param>
       ...
   </X4Config>
   <JavaConfig>
       <Param><Key><!--Java-Option (in startX4.bat)--></Key></Param>
       <!-- weitere <Param/>-Elemente-->
   </JavaConfig>
</Config>
```

Ausgabe

Dieser Funktions-Adapter ergänzt das Input-XML-Dokument sofern möglich um die dynamisch ausgelesenen Werte, beispielsweise:

```
<Config>
   <X4Config>
       <Param>
          <Key><!--Element in X4Config.xml--></Key>
          <Value><!--Konfigurationswert des entsprechenden Elements--></Value>
       </Param>
   </X4Config>
   <JavaConfig>
       <Param>
          <Key><!--Java-Option (in startX4.bat)--></Key>
          <Value><!--Aktueller Wert der Java-Option--></Value>
       </Param>
   </JavaConfig>
</Config>
```

Beispiel

Im Folgenden ein Beispiel für ein ausgegebenes XML-Dokument, das *X4* -Konfigurationsschlüssel- und Werte sowie Java-Optionen und deren Werte enthält:

```
<Config>
    <X4Config>
        <Param>
            <Key>x4.xstore.repositoryPath</Key>
            <Value>C:/X4/X4DB/</Value>
        </Param>
    </X4Config>
    <JavaConfig>
        <Param>
            <Key>x4.config.file</Key>
            <Value>X4config.xml</Value>
        </Param>
        <Param>
            <Key>javax.xml.transform.TransformerFactory</Key>
            <Value>net.sf.saxon.TransformerFactoryImpl</Value>
        </Param>
    </JavaConfig>
</Config>
```

10.22.23 X4 Variable Collector

`X4 Variable Collector`: Sammelt Ergebnisse aus Prozessschleifen und speichert diese in einer Variablen.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • `Collect`: Daten Base64-enkodieren, siehe Abschnitt Operation `Collect` • `Get`: Inhalt der Prozess-Variablen auslesen, siehe Abschnitt Operation `Get` • `GetAndRemove`: Inhalt der Prozess-Variablen auslesen und die Prozess-Variable anschließend entfernen, siehe Abschnitt Operation `GetAndRemove`

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.merge.Collector:* Hauptklasse (Standard)
`name`	Name der Prozess-Variablen
`mode`	Merge-Modus; Bestimmt wie der Input in die Prozess-Variable zusammengeführt wird. **Mögliche Werte:** • `BINARY`: Der Input wird als Byte-Strom zusammengeführt • `TEXT`: Der Input wird als String zusammengeführt • `XML`: Der Input wird als XML-Dokument zusammengeführt

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt
`-1`	Bei der Ausführung der Operation kam es zu technischen Problemen
`0`	Es wurde keine Prozess-Variable gefunden (bei `Get` und `GetAndRemove`)

10.22.23.1 Operation `Collect`

10.22.23.1.1 Modus: `BINARY`

Folgendes gilt für den Modus `BINARY`:

- Adapter-Input wird als Byte-Array interpretiert und an den als Byte-Array interpretierten Inhalt der Prozess-Variablen angehängt.
- `NULL` wird als Byte-Array der Länge `0` betrachtet.
- Der Content-Type des Ergebnisses ist `application/octet-stream`.
- Das Ergebnis wird in die Prozess-Variable zurückgeschrieben und als Adapter-Ergebnis zurückgeliefert.

10.22.23.1.2 Modus: `TEXT`

Folgendes gilt für den Modus `TEXT`:

- Der Adapter-Input wird als String interpretiert (das Encoding wird hierzu ausgewertet) und an den als String interpretierten Inhalt der Prozess-Variablen (hier wird ebenfalls das Encoding verwendet) angehängt.
- `NULL` wird als String der Länge *`0`* betrachtet.
- Der Content-Type des Ergebnisses ist `text/plain` mit dem Encoding `UTF-8`.
- Das Ergebnis wird in die Prozess-Variable zurückgeschrieben und als Adapter-Ergebnis zurückgeliefert.

10.22.23.1.3 Modus: `XML`

Folgendes gilt für den Modus `XML`:

- Der Adapter-Input wird als XML interpretiert und an den als XML interpretierten Inhalt der Prozess-Variablen eingefügt.
- Der Content-Type des Ergebnisses ist `text/xml` und das Encoding `UTF-8`.
- Das Ergebnis wird in die Prozess-Variable zurückgeschrieben und als Adapter-Ergebnis zurückgeliefert.
- Falls der Adapter-Input oder der Inhalt der Prozess-Variablen nicht als XML interpretiert werden kann (`NULL` ist erlaubt), wird der Adapter mit Status *`-1`* und `NULL` als Ausgabe beendet. Der Inhalt der Prozess-Variablen wurde dann nicht verändert.

Angehängt wird wie folgt:

- Es wird ein neues Dokument mit dem Wurzelelement `Merge` erstellt.
- Ist der Inhalt der Prozess-Variablen nicht `NULL`, wird ihr Inhalt wie folgt zusammengeführt.
  Ist das Wurzelelement der Prozess-Variablen `Merge`, werden alle Kind-Element als Kind-Element des Ausgabe-`Merge`-Elements übernommen. Andernfalls wird das Wurzelelement als Kind des Ausgabe-`Merge`-Elements übernommen.
- Anschließend wird dieselbe Behandlung auf den Adapter-Input angewendet.
  Nach einem erfolgreichen `Collect` im XML-Modus befindet sich damit in der Prozess-Variablen

auf jeden Fall ein XML-Dokument mit einem Wurzelelement `Merge`. Aus Sicht der `Collect`-Operation im Modus `XML`, sind `NULL` (Nichts) und das XML-Dokument `<Merge/>` gleichwertig.

10.22.23.2 Operation `Get`

Die Operation `Get` liefert den Inhalt der Prozess-Variablen zurück. Ist die Variable gefüllt, wird der Status *1* zurückgegeben, ist die Variable leer, der Status *0*. Die Operation `Get` verhält sich damit wie die Operation `Test` des Variablen-Bausteins.

Für den Fall, dass die Variable leer war, wird das neutrale Element in die Variable gesetzt und als Adapterausgabe verwendet.

Modus	Entsprechendes neutrales Element
`BINARY`	Byte-Array der Länge *0*
`TEXT`	String der Länge *0*
`XML`	Ein XML-Dokument, das ausschließlich aus dem Wurzelknoten mit dem Namen `Merge` besteht

10.22.23.3 Operation `GetAndRemove`

Die Operation `GetAndRemove` liefert den Inhalt der Prozess-Variablen zurück. Ist die Variable gefüllt, wird der Status *1* zurückgegeben, ist die Variable leer, der Status *0*. Die Operation `GetAndRemove` kombiniert damit die Operationen `Remove` und `Test` des Variablen-Bausteins.

Für den Fall, dass die Variable leer war, wird das neutrale Element in die Variable gesetzt und als Adapterausgabe verwendet.

Modus	Entsprechendes neutrales Element
`BINARY`	Byte-Array der Länge *0*
`TEXT`	String der Länge *0*
`XML`	Ein XML-Dokument, das ausschließlich aus dem Wurzelknoten mit dem Namen `Merge` besteht

Nach dieser Operation ist die Variable leer.

10.23 XML Tools

- Inline Content Expander
- Record Counter
- XHTML Form Extractor
- XML Checker
- XML Element Wrapper
- XML Formatter
- XML Iterator
- XML Text Extractor
- XML Validation
- XPath Iterator

10.23.1 Inline Content Expander

`Inline Content Expander`: Klappt Datensegmente in einem XML-Dokument auf Basis von definierten Parametern ein und aus.

Eigenschaften

`Operation`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** • `Expand`: In einem XML-Dokument eingebettete Datensegmente ausklappen. Die Datensegmente enthalten XML-Daten als reinen Text oder Base64-kodierte Zeichenketten und werden innerhalb des Input-XML-Dokuments ausgeklappt und eingebettet. • `Collapse`: Teile eines XML-Dokuments als reinen Text oder Base64-kodierte Zeichenketten einklappen. Die eingeklappten Datensegmente werden in das Input-XML-Dokument eingebettet.

Parameter

`Adapter`	Main adapter class (do not change!) **Mögliche Werte:** *`de.softproject.integration.adapter.inlinecontentexpanderAdapter`*: Hauptklasse (Standard)
`select`	XPath-Ausdruck zur Auswahl aus den Input-XML-Dokument **Mögliche Werte:** Beliebiger gültiger XPath-Ausdruck (z.B. *`Root/Element[1]`*)
`Format`	Format der Datensegmente **Mögliche Werte:** • `BASE64`: Base64-kodierte Daten (Standard) • `STRING`: Textdaten
`Output`	Ausgabeformat der Datensegmente **Mögliche Werte:** • `XML_ONLY`: Es werden nur gültige XML-Strukturen verarbeitet. Bei ungültigen Strukturen wird ein Fehler ausgegeben. (Standard) • `XML_OR_STRING`: XML-Strukturen werden verarbeitet, wenn sie gültig sind, andernfalls werden die Daten als Zeichenkette ausgegeben. • `STRING_ONLY`: Es werden nur Zeichenketten verarbeitet.

`encoding`	Zeichenkodierung der Datensegmente **Mögliche Werte:** • Beliebige gültige Zeichenkodierung (z.B. *`windows-1252`*) • *`UTF-8`*: UTF-8 (Standard)

Statuswerte

`0`	Der XPath-Ausdruck referenziert kein Element. Das Dokument wird unverändert ausgegeben.
`1`	Der Adapter wurde erfolgreich ausgeführt.
`-1`	Ein Fehler trat bei der Ausführung des Adapters auf. Je nach Operation wird entweder das Dokument wird unverändert ausgegeben (`Collapse`) oder die entsprechende Fehlermeldung ausgegeben (`Expand`).

Input

Der Adapter erwartet ein beliebiges gültiges XML-Dokument als Input.

Beispiel Input-XML-Dokument

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
    <Headers>
        <Header>
            <Name>Content-Type</Name>
            <Value>application/xml</Value>
        </Header>
        <Header>
            <Name>Accept</Name>
            <Value>application/xml</Value>
        </Header>
    </Headers>
    <Body><Content>
            <Description>Adress succesfully submitted!</Description>
            <Data>
                <root>
                    <address>
                        <name>John Doe</name>
                        <street>42 main avenue</street>
                        <zip>13579</zip>
                        <city>example town</city>
                        <state>example state</state>
                        <country>example country</country>
                    </address>
                </root>
            </Data>
        </Content></Body>
</Request>
```

Gesetzte Adapter-Parameter:

- Operation: *Collapse*
- select: *//Body*
- Format: *BASE64*
- Output: *XML_ONLY*
- encoding: *UTF-8*

Ausgabe

Der Adapter gibt ein XML-Dokument mit eingebetteten verarbeiteten Daten aus.

Beispiel-Output

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
    <Headers>
        <Header>
            <Name>Content-Type</Name>
            <Value>application/xml</Value>
        </Header>
        <Header>
            <Name>Accept</Name>
            <Value>application/xml</Value>
        </Header>
    </Headers>
<Body
>PENvbnRlbnQ+DQoJCQk8RGVzY3JpcHRpb24+QWRyZXNzIHN1Y2Nlc2Z1bGx5IHN1Ym1pdHRlZCE8L0Rlc2Ny
aXB0aW9uPg0KCQkJPERhdGE+DQoJCQkJPHJvb3Q+DQoJCQkJCTxhZGRyZXNzPg0KCQkJCQkJPG5hbWU+Sm9ob
iBEb2U8L25hbWU+DQoJCQkJCQk8c3RyZWV0PjQyIG1haW4gYXZlbnVlPC9zdHJlZXQ+DQoJCQkJCQk8emlwPj
EzNTc5PC96aXA+DQoJCQkJCQk8Y2l0eT5leGFtcGxlIHRvd248L2NpdHk+DQoJCQkJCQk8c3RhdGU+ZXhhbXB
sZSBzdGF0ZTwvc3RhdGU+DQoJCQkJCQk8Y291bnRyeT5leGFtcGxlIGNvdW50cnk8L2NvdW50cnk+DQoJCQkJ
CTwvYWRkcmVzcz4NCgkJCQk8L3Jvb3Q+DQoJCQk8L0RhdGE+DQoJCTwvQ29udGVudD4=
</Body>
</Request>
```

10.23.2 Record Counter

`Record Counter`: Zählt Datensätze in den Input-Daten, die anhand eines Trennzeichens bestimmt werden. Dieser Adapter behandelt alle Input-Daten als Text und gibt ein XML-Dokument aus, das die Anzahl der Datensätze enthält.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *Count*: Datensätze in den Input-Daten zählen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.RecordCounter*: Hauptklasse (Standard)
RecordID	Verwendetes Trennzeichen (oder Escape-Sequenz) **Mögliche Werte:** • Beliebige(s) Zeichen (z.B. `,`) • *\n*: Zeilenumbruch als Trennzeichen interpretieren (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Ausgabe

Der erfolgreich ausgeführte *Record Counter* gibt ein XML-Dokument mit folgendem Muster zurück:

```
<RecordCount><!--Anzahl der Datensätze--></RecordCount>
```

10.23.3 XHTML Form Extractor

`XHTML Form Extractor`: Extrahiert Formularfelder aus XML-kompatiblen (X)HTML-Dokumenten, um das Mapping von Prozessdaten auf Formularfelder übersichtlicher zu gestalten.
Der Adapter erwartet als Input ein (X)HTML-Dokument (in jedem Fall ein wohlgeformtes XML-Dokument!) und gibt ein XML-Dokument mit der Struktur des extrahierten (X)HTML-Formulars aus. Zudem kann der Adapter über Operation `FillInFields` die adapterspezifische XML-Struktur, die gesetzte Formularfelder und ggf. veränderte Eigenschaften für Optionsfelder enthält, in eine (X)HTML-Formularvorlage einfüllen und das fertige (X)HTML-Formulardokument ausgeben.

Dieser Adapter richtet sich an Entwickler von *Human Workflow*-Prozessen, die bestehende (X)HTML-Formularvorlagen zur Laufzeit mit Daten befüllen möchten. Hierzu wird in einem XSL-Mapping die Input-Datenstruktur (z. B. die Ergebnisse einer Datenbankabfrage) über den Mapping Editor mit den Feldern des (X)HTML-Formulars verknüpft.
Um (X)HTML-Dokumentstrukturen für das Mapping übersichtlicher zu gestalten, extrahiert der `XHTML Form Extractor Adapter` alle Formularelemente, sodass im Mapping Editor nur diese vereinfachte Formularstruktur und die Daten verknüpft werden müssen. Zur Konvertierung in die XML-Formularstruktur ist im *X4 Designer* die Kontextmenüfunktion **Extract XHTML Form** für (X)HTML-Repository-Dokumente verfügbar, die diesen Adapter nutzt.
Um die XML-Formularstruktur nach dem Mapping wieder in ein vollständiges (X)HTML-Formular umzuwandeln, den `XHTML Form Extractor Adapter` als nächsten Prozess-Schritt platzieren, die Operation `FillInFields` wählen und in Adapter-Parameter `templateUrl` die URL zur (X)HTML-Formularvorlage angeben. Bei der Ausführung des Prozesses werden somit die veränderten Formularelemente mit der (X)HTML-Formularvorlage zusammengeführt und zur Laufzeit ausgegeben, z. B. zur Anzeige in Web-Anwendungen. Über Attribute in der XML-Formularstruktur lassen sich Optionsfelder (de)selektieren und (de)aktivieren sowie CSS-Direktformatierungen und CSS-Klassenzuordnungen definieren.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`ExtractFields`*: Formularfelder aus einem Input-(X)HTML-Dokument (wohlgeformtes XML!), extrahieren • *`FillInFields`*: Formularfelder aus der adapterspezifischen XML-Struktur in ein XML-konformes (X)HTML-Dokument einfügen, das in Parameter `templateUrl` definiert wurde; zusätzliche Felder aus dem (X)HTML-Vorlagendokument werden unverändert übernommen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** de.softproject.integration.adapter.htmlformextractor.FormAdapter: Hauptklasse (Standard)
templateUrl	URL zum Vorlagendokument des (X)HTML-Formulars (benötigt für Operation `FillInFields`) **Mögliche Werte:** Beliebige gültige URL (z.B. `xstore://Project/Folder/Form.xhtml`)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

Beispiel

In diesem Beispiel dient ein HTML5-Formular (wohlgeformtes XML) als Input:

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My Form</title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
    <style type="text/css">
      body { background-color:#F0F0F0; font-family:sans-serif }
    </style>
  </head>
  <body>
    <h1>Form</h1>
    <form action="ACTION" method="post" id="myID">
      <p>
      <input type="text" name="myName" value=""/>
        <select name="myDropdown" size="1">
          <option>1</option>
          <option>2</option>
          <option>3</option>
        </select>
      <input type="radio" name="myOptions" value="either" /> Either
        <input type="radio" name="myOptions" value="or" /> Or
      <input type="submit" value="submit" name="mySend"/>
     </p>
    </form>
  </body>
</html>
```

Bei Operation `ExtractFields` wandelt der Adapter das HTML5-Formular in folgende XML-Struktur um:

```
<HtmlFields>
  <title>My Form</title>
  <form_myID id="myID">
   <text_myName name="myName" type="text"/>
   <submit_mySend name="mySend" type="submit">submit</submit_mySend>
   <singleselect_myDropdown name="myDropdown">
     <option disabled="false" selected="false">1</option>
     <option disabled="false" selected="false">2</option>
     <option disabled="false" selected="false">3</option>
   </singleselect_myDropdown>
  </form_myID>
</HtmlFields>
```

Über ein XSL-Mapping wurde nun folgende XML-Struktur erzeugt, in der Feldwerte und der Seitentitel geändert sowie verschiedene Eigenschaften für Optionsfelder gesetzt sind und ein zusätzliches Optionsfeld ergänzt wurde.

```
<HtmlFields>
  <title>X4 Experience</title>
  <form_myID id="myID">
   <text_myName type="text" name="myName">X4 Experience</text_myName>
   <submit_mySend type="submit" name="mySend">submit</submit_mySend>
   <singleselect_myDropdown name="myDropdown">
     <option>Version 4.0</option>
     <option selected="false" disabled="true">Version 4.1</option>
     <option selected="false" disabled="false" cssStyle="color:blue">Version 4.2
     </option>
     <option selected="true" disabled="false">Version 4.3</option>
   </singleselect_myDropdown>
  </form_myID>
</HtmlFields>
```

Mit Operation `FillInFields` wandelt der Adapter die oben genannte modifizierte XML-Struktur wieder in ein vollständiges, XML-konformes HTML5-Formular um:

```
<!DOCTYPE HTML>
<html>
  <head>
   <title>X4 Experience</title>
   <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
   <style type="text/css">
     body { background-color:#F0F0F0; font-family:sans-serif }
   </style>
  </head>
  <body>
   <h1>Form</h1>
   <form action="ACTION" method="post" id="myID">
     <p>
      <input type="text" name="myName" value="X4 Experience"
       style="color:blue"/>
      <select name="myDropdown" size="1">
        <option>Version 4.0</option>
        <option disabled="disabled">Version 4.1</option>
        <option style="color:blue">Version 4.2</option>
        <option selected="selected">Version 4.3</option>
      </select>
      <input type="radio" name="myOptions" value="either"/> Either
      <input type="radio" name="myOptions" value="or"/> Or
      <input type="submit" value="submit" name="mySend"/>
    </p>
   </form>
  </body>
</html>
```

10.23.3.1 Operation: ExtractFields

Mit Operation `ExtractFields` lassen sich Formularfelder aus einem Input-(X)HTML-Dokument (wohlgeformtes XML!), extrahieren.

Input

Bei Operation `ExtractFields` erwartet der Adapter ein (X)HTML-Dokument, das eine wohlgeformte XML-Struktur aufweist (korrekte Verschachtelung, selbstschließende leere Elemente wie `<br/>` etc.) und dessen Formularelemente über Attribut `id` oder `name` eindeutig sind.

Ausgabe

Bei Operation `ExtractFields` gibt der Adapter ein XML-Dokument mit Formulardaten aus, das Sie im Mapping als Vorlage für die Zielstruktur verwenden können.

10.23.3.2 Operation: FillInFields

Mit Operation `FillInFields` lassen sich Formularfelder aus der adapterspezifischen XML-Struktur in ein XML-konformes (X)HTML-Dokument einfügen, das in Parameter `templateUrl` definiert wurde; zusätzliche Felder aus dem (X)HTML-Vorlagendokument werden unverändert übernommen.

Input

Bei Operation `FillInFields` erwartet der Adapter ein XML-Dokument, das eine Struktur nach folgendem Muster aufweist und für alle Formularfelder (außer Checkboxen und Radio-Buttons) ein XML-Element erzeugt, das Feldtyp und Feldname im Namen enthält.

In der XML-Struktur können `<option/>`-Felder über spezifische Attribute mit CSS-Direktformatierungen (`cssStyle="..."`), CSS-Klassenzuordnungen (`cssClass="..."`) und Labels (`label="..."`) versehen sowie (de)selektiert (`selected="true/false"`), (de)aktiviert (`disabled="true/false"`) und auf nur lesend gesetzt (`readonly="true/false"`) werden. Werte von `<input/>` und `<option/>` lassen sich über ein Attribut `value="..."` setzen.

```
<HtmlFields>
 <title><!--Seitentitel des (X)HTML-Dokuments--></title>
 <!-- Beliebig viele Formulare (eindeutig über ID identifiziert) -->
 <form_id id="Formular-ID">
  <!-- Beliebig viele Formularfelder (identifiziert über ID oder Name) -->
  <Feldtyp_Feldname/-ID name="Feldname" type="Feldtyp"><!--Feldwert--></
Feldtyp_Feldname/-ID>
 </form_id>
</HtmlFields>
```

Ausgabe

Bei Operation `FillInFields` gibt der Adapter ein XML-kompatibles (X)HTML-Dokument aus, das die Feldwerte (und bei Optionsfeldern ggf. geänderte Eigenschaften) enthält, die im Input-Dokument

definiert wurden und dessen Struktur der in Parameter `templateUrl` definierten Vorlage entspricht. Checkboxen und Radio-Buttons aus der Vorlage werden unverändert in das Ergebnisdokument übernommen.

10.23.4 XML Checker

`XML Checker`: Prüft, ob das Input-Dokument ein XML-Dokument ist, und gibt das Dokument sowie einen entsprechenden Status aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *`checkXML`*: Input-Dokument prüfen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.typechecker.TypeCheckerAdapter`*: Hauptklasse (Standard)

Statuswerte

0	Das geprüfte Input-Dokument ist kein XML-Dokument.
1	Das geprüfte Input-Dokument ist ein wohlgeformtes XML-Dokument.

10.23.5 XML Element Wrapper

`XML Element Wrapper`: Umschließt die Input-Daten mit einem bestimmten XML-Wurzelelement und gibt ein UTF-8-kodiertes XML-Dokument aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** *`Wrap`*: Input-Daten als XML-Dokument mit Wurzelelement aus Parameter `name` ausgeben

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.elementwrapper.ElementWrapperAdapter`*: Hauptklasse (Standard)

name	Name des Wurzelelements **Mögliche Werte:** • Beliebiger gültiger XML-Elementname • *`Data`*: Wurzelelement *`Data`* (Standard)

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.23.6 XML Formatter

`XML Formatter`: Stellt Formatierungsfunktionen für XML-Dokumente bereit, z.B. Zeilenumbrüche und Einrückungen.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`PrettyPrint`*: XML-Dokument formatieren

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.xmlformatter.XMLFormatterAdapter*: Hauptklasse (Standard)
indentSize	Anzahl der Einrückungen innerhalb des XML-Dokuments **Mögliche Werte:** Beliebige ganze Zahl, z. B. *4*

Statuswerte

1 (successful)	Die Anfrage war erfolgreich und ein Ergebnis wurde zurückgegeben
-1 (failed)	Die Anfrage schlug aufgrund eines technischen Fehlers fehl

Input

Der Adapter erwartet als Input ein unformatiertes XML-Dokument.

Ausgabe

Der Adapter gibt ein formatiertes XML-Dokument zurück.

10.23.7 XML Iterator

`XML Iterator`: Liest ein XML-Dokument von einer beliebigen gültigen URL schrittweise in einen Technical Process ein und gibt bei jeder Iteration ein Kindelement (inklusive sämtlicher Kindknoten) des Wurzel-Elements als XML-Dokument aus.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`Iterate`*: Kindknoten des Wurzel-Elements einlesen • *`Clear`*: Iterator eines *XML Iterators* zurücksetzen, der dieselbe ID besitzt wie der gewünschte *XML Iterator* mit der Operation *`Iterate`* (siehe Parameter `id`) ⚠ Nach Aufruf der *`Iterate`* Operation sollten Sie die *`Clear`* Operation aufrufen, um Ressourcenlecks zu vermeiden.

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.xmliterator.XmlIterator`*: Hauptklasse (Standard)
url	URL zum einzulesenden XML-Dokument **Mögliche Werte:** • Beliebige gültige URL (z.B. *`xstore://Project/Folder/File.xml`*) • URL zu einem UNC-Netzlaufwerk (z.B. *`file://///storage/share_data/Folder/File.xml`*)

id	Eindeutige ID des *XML Iterator*-Bausteins innerhalb des Prozess-Kontextes bzw. der Prozess-Instanz (wenn Sie mehrere voneinander unabhängig iterierende *XML Iterator Adapter* in demselben Prozess-Kontext verwenden möchten) **Mögliche Werte:** • *0* : Aktuelle Action-ID (Nummer des Prozess-Schritts des *XML Iterator Adapters* im aktuellen Prozess) als ID verwenden; Beachten Sie hierbei, dass ein *XML Iterator Adapter* in einem Subprozess die gleiche Action-ID besitzen könnte (Standard) • Beliebige ganze Zahl außer *0*, um den jeweiligen *XML Iterator*-Baustein innerhalb des Prozess-Kontextes eindeutig zu kennzeichnen; Empfehlung: einen Wert *>1000* verwenden, um nicht zufällig eine bereits bestehende Action-ID zu "erwischen"

Statuswerte

-1 (error)	Fehler bei der Ausführung des Adapters (Details siehe Server-Protokoll)
0 (empty)	Der Adapter gibt keine Daten (mehr) aus. Dieser Status kann als Abbruchbedingung einer Schleife im Technical Process ausgewertet werden und gibt an, dass keine weiteren Datensätze verfügbar sind.
1 (successful)	Der Adapter gibt ein Ergebnis aus. Dieser Status kann als Schleifeneintrittsbedingung im Technical Process ausgewertet werden und gibt an, dass ein weiterer Datensatz verfügbar ist.

10.23.8 XML Text Extractor

`XML Text Extractor`: Extrahiert den Inhalt eines Knotens innerhalb eines XML-Dokuments als Text-Dokument. Der Knoten wird mithilfe eines XPath-Ausdrucks angesteuert.

> **Extrahierbare Elemente**
> Sie können nur Elemente des Typs CDATA, Attribute oder Text extrahieren.

Eigenschaften

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • `Extract`: Inhalt eines Knotens extrahieren

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.xmltextextractor.XMLTextExtractor`*: Hauptklasse (Standard)
`xpath`	XPath-Ausdruck zur Auswahl aus dem Input-XML-Dokument **Mögliche Werte:** Beliebiger gültiger XPath-Ausdruck (z.B. *`Root/Element[1]`*) Wenn der XPath-Ausdruck mehr als ein Element ansteuert, wird ein Fehler ausgegeben. Um mehr als ein Element zu extrahieren, kombinieren Sie den Adapter mit dem XPath Iterator-Adapter.

Statuswerte

`0`	Der Adapter gibt keine Daten (mehr) aus. Dieser Status kann als Abbruchbedingung einer Schleife im Technical Process ausgewertet werden und gibt an, dass keine weiteren Datensätze verfügbar sind.
`1`	Der Adapter gibt ein Ergebnis aus. Dieser Status kann als Schleifeneintrittsbedingung im Technical Process ausgewertet werden und gibt an, dass ein weiterer Datensatz verfügbar ist.
`-1`	Fehler bei der Ausführung des Adapters (Details siehe Server-Protokoll)

Input

Der Adapter erwartet ein beliebiges gültiges XML-Dokument als Input.

Beispiel Input-XML-Dokument

```
<RootElement>
    <address>
        <name>John Doe</name>
        <street>42 Main Avenue</street>
        <zip>13579</zip>
        <city>Example town</city>
        <state>Example state</state>
        <country>Example country</country>
    </address>
    <address>
        <name>Jane Doe</name>
        <street>22nd Street</street>
        <zip>84643</zip>
        <city>Example city</city>
        <state>Example state</state>
        <country>Example country</country>
    </address>
</RootElement>
```

Gesetzte Adapter-Parameter:

- `xpath`: *`//address[1]/name`*

Ausgabe

Der Adapter gibt ein Text-Dokument mit verarbeiteten Daten aus.

Beispiel-Output

```
John Doe
```

10.23.9 XML Validation

`XML Validation`: Validiert ein Input-XML-Dokument gegen ein XML-Schema oder ein *Relax NG*-Schema.
Ist das geprüfte XML-Dokument gültig, gibt der Adapter dieses unverändert aus; Wenn das geprüfte XML-Dokument ungültig ist, wird ein XML-Dokument mit einer entsprechenden Fehlermeldung ausgegeben.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** `Validate`: XML-Dokument validieren

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.xmlvalidator.XMLValidator`*: Hauptklasse (Standard)
`SchemaUrl`	Pfad zur Schema-Definition (XML-Schema oder Relax NG) **Mögliche Werte:** Beliebige `xstore://`-URL (z. B. *`xstore://Project/Folder/Schema.xsd`*)
`Verbose`	Angabe, ob das Ausgabedokument detaillierte Informationen zum Validierungssfehler ausgeben soll **Mögliche Werte:** • `true`: Das Ausgabedokument enthält detaillierte Informationen zum Validierungsfehler (Standard) • `false`: Der Adapter beendet die Validierung nach dem ersten Fehler und gibt keine Informationen über die Fehlerursache aus
`xsdVersion`	XSD-Version, die zum Validieren der XML-Dokumente verwendet werden soll **Mögliche Werte:** • `1.0`: XSD-Version 1.0 wird verwendet • `1.1`: XSD-Version 1.1 wird verwendet

Statuswerte

`-1 (error)`	Während der Ausführung des Adapters trat ein Fehler auf. Nähere Details finden Sie im Server-Log.
`1 (successful)`	Das geprüfte XML-Dokument ist gültig gemäß dem angegebenen Schema; Der Adapter gab das geprüfte Input-XML-Dokument unverändert aus.
`2`	Das geprüfte XML-Dokument ist nicht gültig gemäß dem angegebenen Schema; Der Adapter gab eine Fehlermeldung aus (siehe "Output").

Ausgabe

Dieser Adapter gibt ein UTF-8-kodiertes XML-Dokument aus. War die Gültigkeitsprüfung erfolgreich, wird das Input-Dokument unverändert ausgegeben. Ist das geprüfte XML-Dokument ungültig, wird ein XML-Dokument mit einer entsprechenden Fehlermeldung ausgegeben, wobei für jeden Fehler ein XML-Element `<Error/>` erzeugt wird, das eine detaillierte Fehlerbeschreibung enthält, zum Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?>
<ValidationErrors>
    <Error>cvc-maxLength-valid: Value 'abc123' with length = '6' is not facet-valid
with respect to maxLength '5' for type 'id'.</Error>
    <Error>cvc-type.3.1.3: The value 'abc123' of element 'Id' is not valid.</Error>
</ValidationErrors>
```

10.23.10 XPath Iterator

`XPath Iterator`: Selektiert eine Knotengruppe, einen Knoten oder Knoteninhalt über einen XPath-Ausdruck in einem Input-XML-Dokument und gibt je nach Ausdruck eine Knotengruppe als XML-Dokument oder die Werte als Text-Dokument aus. Der Adapter kann dabei alle zutreffenden Knoten durchlaufen und die Ergebnisse nacheinander ausgeben (`ForEach`-Schleife) oder das Ergebnis des ersten Treffers ausgeben.

> Wenn Sie innerhalb eines Prozess-Kontextes mehrere *XPath Iteratoren* definieren, dann vergeben Sie in Parameter `Id` eine eindeutige ID für den jeweiligen Adapter, z.B. *`1001`*, *`1002`* etc. So wird für das Selektieren im `forEach`-Modus und für das Zurücksetzen eines Iterators gewährleistet, dass nur der *XPath Iterator* mit der entsprechenden ID angesprochen wird (siehe Operation *`Clear`*).
> Wird in Parameter `Id` der Standardwert *`0`* belassen, dann wird die Nummer des Prozess-Schritts (Action ID) im aktuellen Prozess als ID verwendet. Diese ID ist möglicherweise nicht eindeutig, wenn sich in einem Subprozess ein *XPath Iterator* mit der gleichen Action-ID befindet.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`Select`*: XPath-Ausdruck auf das Input-XML-Dokument anwenden und Ergebnis-Dokument ausgeben (Parameter `Id` beachten) • *`Clear`*: Iterator eines *XPath Iterators* zurücksetzen, der dieselbe ID besitzt wie der gewünschte *XPath Iterator* mit der Operation *`Select`* (siehe Parameter `Id`)

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.xpathselector.XPathSelector`*: Hauptklasse (Standard)

XPath	XPath-Ausdruck, der auf das Input-XML-Dokument angewandt wird **Mögliche Werte:** Beliebiger XPath-Ausdruck (z.B. `/Element[@att="Value"]/text()`)
forEach	Den ersten oder alle zutreffenden Knoten selektieren **Mögliche Werte:** • *ja*: Alle selektierten Knoten nacheinander durchlaufen und jeweils ein Dokument ausgeben (nützlich in Kombination mit einem Condition-Prozessbaustein) (Standard) • *nein*: Ersten Treffer ausgeben
Id	Eindeutige ID des *XPath Iterator*-Bausteins innerhalb des Prozess-Kontextes bzw. der Prozess-Instanz (wenn Sie mehrere voneinander unabhängig iterierende *XPath Iterator Adapter* in demselben Prozess-Kontext verwenden möchten) **Mögliche Werte:** • `0`: Aktuelle Action-ID (Nummer des Prozess-Schritts des *XPath Iterators* im aktuellen Prozess) als ID verwenden; Beachten Sie hierbei, dass ein *XPath Iterator* in einem Subprozess die gleiche Action-ID besitzen könnte (Standard) • Beliebige ganze Zahl außer `0`, um den jeweiligen *XPath Iterator*-Baustein innerhalb des Prozess-Kontextes eindeutig zu kennzeichnen; Empfehlung: einen Wert `>1000` verwenden, um nicht zufällig eine bereits bestehende Action-ID zu "erwischen"

Statuswerte

-1 (error)	Fehler bei der Ausführung des Adapters (Details siehe Server-Log)
0 (empty)	Der Adapter gibt keine Daten (mehr) aus. Dieser Status kann im Iterator-Modus als Abbruchbedingung einer Schleife im Technical Process ausgewertet werden und gibt an, dass keine weiteren Datensätze verfügbar sind.
1 (successful)	Der Adapter gibt ein Ergebnis aus. Dieser Status kann im Iterator-Modus als Schleifeneintrittsbedingung im Technical Process ausgewertet werden und gibt an, dass ein weiterer Datensatz verfügbar ist.

10.24 Veraltete Adapter

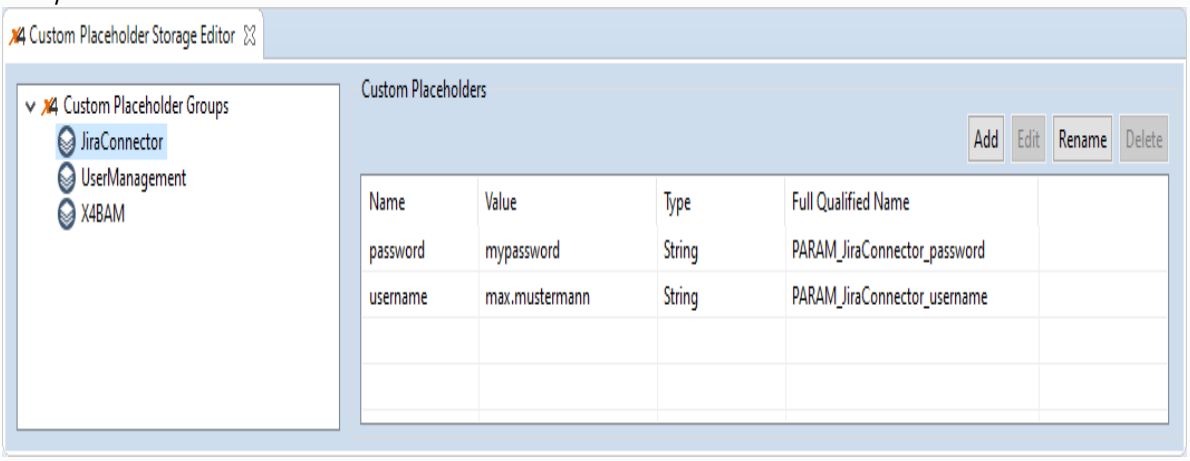
In diesem Abschnitt sind alle veralteten Adapter aufgelistet. Veraltete Adapter werden nicht aktualisiert oder angepasst.

10.24.1 Atlassian Jira Connector

`Atlassian Jira Connector`: Verbindet Prozesse mit Atlassian Jira.

Beachten Sie, dass Sie für die Nutzung des Adapters zunächst eine Parametergruppe mit den entsprechenden Jira-Zugangsdaten anlegen müssen. Diese können Sie entweder manuell in der Global Parameter Configuration oder über einen Integration Process mithilfe des Adapter Global Parameter Management.

Beispiel:



Properties (Eigenschaften)

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • `Get Issue`: Vorgang aus einem bestimmten Projekt aufrufen • `Get all Issues`: Alle Vorgänge aus einem bestimmten Projekt aufrufen • `Create Issue`: Vorgang in einem bestimmten Projekt anlegen • `Edit Issue`: Verfügbare Felder in einem bestehenden Vorgang bearbeiten • `Add Attachment`: Anhang zu einem bestimmten Vorgang hinzufügen • `Remove Attachment`: Anhang eines Vorgangs anhand der globalen ID löschen • `Get all Comments`: Alle Kommentare eines bestimmten Vorgangs abrufen • `Get Comment`: Einen bestimmten Kommentar aus einem Vorgang abrufen • `Add Comment`: Einen neuen Kommentar zu einem Vorgang hinzufügen • `Edit Comment`: Einen bestimmten Kommentar bearbeiten • `Search`: Eine Suche anhand von Parametern durchführen • `Execute Transition`: Statusübergang für einen bestimmten Vorgang ausführen • `Get Transition`: Status für einen Vorgang ausgeben

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.nativeadapter.NativeAdapter`* : Hauptklasse (Standard)
`Repository`	Virtuelles Repository des Adapter-Projekts. Der Wert kann nicht geändert werden.
`host`	URL des Jira-Servers **Mögliche Werte: G**ültige URL, z. B. *`jira.example.com`*

`port`	Port der ReST-API **Mögliche Werte:** • Positive ganze Zahl • *443* : Standardport
`key`	Projektschlüssel mit oder ohne Vorgangs-ID **Mögliche Werte:** Gültiger Projektschlüssel, z. B. `XVIER`, *`XVIER-12345`*
`secure`	Angabe, ob die Kommunikation über HTTPS stattfinden soll oder nicht **Mögliche Werte:** `true` (Standard) / `false`
`username`	Jira-Benutzer, der für die Authentifizierung verwendet wird
`password`	Passwort des Jira-Benutzers für die Authentifizierung
`token`	Für die Authentifizierung verwendetes Token
`query`	*Nutzung in Verbindung mit der Operation `Get all Issues`;* Suchkriterium als "Jira Query Language"-Ausdruck. **Mögliche Werte: G**ültiges Suchkriterium, z. B. • `project=XVIER and Rang < 999` • `project=ITSERVICE and reporter=max.mustermann`
`offset`	Offset, bei dem die Aufzählung der Vorgänge beginnen soll
`count`	Maximale Anzahl der zurückgelieferten Vorgänge
`id`	ID zur Spezifizierung von Kommentaren oder Anhängen

Statuswerte

1	Die Operation wurde erfolgreich ausgeführt.
-1	Es kam zu einem Fehler während der Ausführung.

Input

Der Adapter erwartet verschiedene Input-Strukturen abhängig von der ausgewählten Operation. Weitere Informationen finden Sie in den folgenden Unterkapiteln.

Output

Der Adapter gibt verschiedene Strukturen abhängig von der gewählten Operation aus. Weitere Informationen finden Sie in den folgenden Unterkapiteln.

10.24.1.1 Get Issue

Input

- Für diese Operation wird kein Input benötigt.
- Für diese Operation sind die Parameter `host`, `key`, `username` und `password` oder `token` erforderlich.
- Der Projektschlüssel mit der Vorgangs-ID wird im Parameter `key` angegeben.

Output

- Wenn die Operation erfolgreich ausgeführt wurde, dann gibt der Adapter den Vorgang mit allen ausgefüllten Feldern aus.

Get Issue

```
<JSON>
   <expand>renderedFields,names,schema,operations,editmeta,changelog,versionedRepresentations</expand>
   <id>87606</id>
   <self>https://jira.softproject.de/rest/api/2/issue/87606</self>
   <key>XVIERSC-174</key>
   <fields>
      <customfield_12130/>
      <customfield_12132/>
      <customfield_12131/>
      <fixVersions/>
      <customfield_12133/>
      <customfield_12411/>
      <resolution/>
      <customfield_12410/>
      <customfield_12413/>
      <customfield_12412/>
      <customfield_12129/>
      <customfield_11711/>
      <customfield_12404/>
      <customfield_12128/>
      <customfield_11710/>
      <customfield_12403/>
      <customfield_11713/>
      <customfield_12406/>
      <customfield_11712/>
      <customfield_12405/>
      <customfield_11715/>
      <customfield_11714/>
      <customfield_12407/>
      <customfield_11719/>
      <lastViewed/>
      <customfield_12000/>
      <customfield_12121/>
      <customfield_12120/>
      <customfield_12123/>
      <customfield_12002/>
      <customfield_12001/>
      <customfield_12122/>
      <priority>
         <self>https://jira.softproject.de/rest/api/2/priority/3</self>
         <iconUrl>https://jira.softproject.de/images/icons/priorities/major.svg</iconUrl>
         <name>Major</name>
         <id>3</id>
      </priority>
      <customfield_12400/>
      <customfield_12124/>
      <customfield_12127/>
      <labels/>
      <customfield_12401/>
```

```
      <customfield_11700/>
      <customfield_12118/>
      <customfield_12514/>
      <customfield_12513/>
      <customfield_11702/>
      <customfield_11701/>
      <customfield_12119/>
      <customfield_11704/>
      <aggregatetimeoriginalestimate/>
      <timeestimate/>
      <versions/>
      <customfield_11703/>
      <customfield_11706/>
      <customfield_11705/>
      <customfield_11708/>
      <customfield_11707/>
      <issuelinks/>
      <customfield_11709/>
      <assignee/>
      <status>
         <self>https://jira.softproject.de/rest/api/2/status/10206</self>
         <description>Noch nicht abgeschätzte und eingeplante Aufgaben</description>
         <iconUrl>https://jira.softproject.de/images/icons/statuses/generic.png</
iconUrl>
         <name>New</name>
         <id>10206</id>
         <statusCategory>
            <self>https://jira.softproject.de/rest/api/2/statuscategory/2</self>
            <id>2</id>
            <key>new</key>
            <colorName>blue-gray</colorName>
            <name>To Do</name>
         </statusCategory>
      </status>
      <components>
         <e>
            <self>https://jira.softproject.de/rest/api/2/component/20009</self>
            <id>20009</id>
            <name>Demos</name>
         </e>
      </components>
      <customfield_12110/>
      <customfield_12112/>
      <customfield_12111/>
      <customfield_12114/>
      <customfield_12510>
         <self>https://jira.softproject.de/rest/api/2/customFieldOption/11918</self>
         <value>Ja</value>
         <id>11918</id>
      </customfield_12510>
      <customfield_12113/>
      <customfield_11300/>
      <customfield_12116/>
      <customfield_12512/>
      <customfield_12115/>
```

```
      <customfield_12511/>
      <customfield_11810/>
      <customfield_12503/>
      <customfield_12502/>
      <customfield_11812/>
      <customfield_12505/>
      <customfield_11811/>
      <customfield_12507/>
      <aggregatetimeestimate/>
      <customfield_12506/>
      <customfield_12508/>
      <creator>
         <self>https://jira.softproject.de/rest/api/2/user?
username=x4solution.center</self>
         <name>x4solution.center</name>
         <key>JIRAUSER12810</key>
         <emailAddress>x4solution.center@softproject.de</emailAddress>
         <avatarUrls>
            <_48x48>https://jira.softproject.de/secure/useravatar?
ownerId=JIRAUSER12810&avatarId=14718</_48x48>
            <_24x24>https://jira.softproject.de/secure/useravatar?
size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
            <_16x16>https://jira.softproject.de/secure/useravatar?
size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
            <_32x32>https://jira.softproject.de/secure/useravatar?
size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
         </avatarUrls>
         <displayName>X4 Solution Center</displayName>
         <active>true</active>
         <timeZone>Europe/Berlin</timeZone>
      </creator>
      <subtasks/>
      <reporter>
         <self>https://jira.softproject.de/rest/api/2/user?
username=x4solution.center</self>
         <name>x4solution.center</name>
         <key>JIRAUSER12810</key>
         <emailAddress>x4solution.center@softproject.de</emailAddress>
         <avatarUrls>
            <_48x48>https://jira.softproject.de/secure/useravatar?
ownerId=JIRAUSER12810&avatarId=14718</_48x48>
            <_24x24>https://jira.softproject.de/secure/useravatar?
size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
            <_16x16>https://jira.softproject.de/secure/useravatar?
size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
            <_32x32>https://jira.softproject.de/secure/useravatar?
size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
         </avatarUrls>
         <displayName>X4 Solution Center</displayName>
         <active>true</active>
         <timeZone>Europe/Berlin</timeZone>
      </reporter>
      <aggregateprogress>
         <progress>0</progress>
         <total>0</total>
```

```
      </aggregateprogress>
      <customfield_12500/>
      <customfield_12612/>
      <customfield_12219/>
      <customfield_11801/>
      <customfield_11800/>
      <customfield_11803/>
      <customfield_11802/>
      <customfield_11805/>
      <customfield_11804/>
      <customfield_11807/>
      <customfield_11806/>
      <progress>
         <progress>0</progress>
         <total>0</total>
      </progress>
      <customfield_11809/>
      <votes>
         <self>https://jira.softproject.de/rest/api/2/issue/XVIERSC-174/votes</self>
         <votes>0</votes>
         <hasVoted>false</hasVoted>
      </votes>
      <worklog>
         <startAt>0</startAt>
         <maxResults>20</maxResults>
         <total>0</total>
         <worklogs/>
      </worklog>
      <customfield_11808/>
      <issuetype>
         <self>https://jira.softproject.de/rest/api/2/issuetype/7</self>
         <id>7</id>
         <description>Created by Jira Agile - do not edit or delete. Issue type for a
user story.</description>
         <iconUrl>https://jira.softproject.de/secure/viewavatar?
size=xsmall&avatarId=11815&avatarType=issuetype</iconUrl>
         <name>Story</name>
         <subtask>false</subtask>
         <avatarId>11815</avatarId>
      </issuetype>
      <timespent/>
      <project>
         <self>https://jira.softproject.de/rest/api/2/project/14505</self>
         <id>14505</id>
         <key>XVIERSC</key>
         <name>X4 Solution Center</name>
         <projectTypeKey>software</projectTypeKey>
         <avatarUrls>
            <_48x48>https://jira.softproject.de/secure/projectavatar?
pid=14505&avatarId=14707</_48x48>
            <_24x24>https://jira.softproject.de/secure/projectavatar?
size=small&pid=14505&avatarId=14707</_24x24>
            <_16x16>https://jira.softproject.de/secure/projectavatar?
size=xsmall&pid=14505&avatarId=14707</_16x16>
```

```
            <_32x32>https://jira.softproject.de/secure/projectavatar?
size=medium&pid=14505&avatarId=14707</_32x32>
         </avatarUrls>
         <projectCategory>
            <self>https://jira.softproject.de/rest/api/2/projectCategory/10000</self>
            <id>10000</id>
            <description>SoftProject Produkte</description>
            <name>Produkt</name>
         </projectCategory>
      </project>
      <customfield_11001>
         <self>https://jira.softproject.de/rest/api/2/customFieldOption/10303</self>
         <value>-1 - Unbekannt / Nicht bewertet</value>
         <id>10303</id>
      </customfield_11001>
      <customfield_11002>987.0</customfield_11002>
      <aggregatetimespent/>
      <customfield_12611/>
      <customfield_12610>
         <self>https://jira.softproject.de/rest/api/2/customFieldOption/12049</self>
         <value>Not deployed</value>
         <id>12049</id>
      </customfield_12610>
      <customfield_11400/>
      <customfield_12601/>
      <customfield_10700/>
      <customfield_12604/>
      <customfield_12603/>
      <customfield_12606/>
      <customfield_12605/>
      <customfield_12608/>
      <resolutiondate/>
      <customfield_12607/>
      <customfield_12609/>
      <workratio>-1</workratio>
      <watches>
         <self>https://jira.softproject.de/rest/api/2/issue/XVIERSC-174/watchers</
self>
         <watchCount>1</watchCount>
         <isWatching>true</isWatching>
      </watches>
      <created>2021-02-05T13:28:47.218+0100</created>
      <customfield_12200>
         <self>https://jira.softproject.de/rest/api/2/customFieldOption/11403</self>
         <value>Störung</value>
         <id>11403</id>
      </customfield_12200>
      <customfield_12600/>
      <customfield_10139/>
      <updated>2021-02-05T13:35:38.823+0100</updated>
      <timeoriginalestimate/>
      <description>This issue is created using the Atlassian Jira Connector with a
REST API</description>
      <customfield_11100>
         <self>https://jira.softproject.de/rest/api/2/customFieldOption/10400</self>
```

```
         <value>Produkt</value>
         <id>10400</id>
      </customfield_11100>
      <customfield_11101>
         <self>https://jira.softproject.de/rest/api/2/customFieldOption/10404</self>
         <value>-1 - Unbekannt / Nicht bewertet</value>
         <id>10404</id>
      </customfield_11101>
      <customfield_11740/>
      <customfield_11500>{summaryBean=com.atlassian.jira.plugin.devstatus.rest.Summar
yBean@405cbc05[summary={pullrequest=com.atlassian.jira.plugin.devstatus.rest.SummaryI
temBean@243f9af7[overall=PullRequestOverallBean{stateCount=0, state='OPEN',
details=PullRequestOverallDetails{openCount=0, mergedCount=0,
declinedCount=0}},byInstanceType={}],
build=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@4a6a154[overall=com.at
lassian.jira.plugin.devstatus.summary.beans.BuildOverallBean@5c966b15[failedBuildCoun
t=0,successfulBuildCount=0,unknownBuildCount=0,count=0,lastUpdated=<null
>,lastUpdatedTimestamp=<null>],byInstanceType={}],
review=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@4eb90470[overall=com.
atlassian.jira.plugin.devstatus.summary.beans.ReviewsOverallBean@13b44058[stateCount=
0,state=<null>,dueDate=<null>,overDue=false,count=0,lastUpdated=<null
>,lastUpdatedTimestamp=<null>],byInstanceType={}], deployment-
environment=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@2229ff7b[overall
=com.atlassian.jira.plugin.devstatus.summary.beans.DeploymentOverallBean@4d88e6e8[top
Environments=[],showProjects=false,successfulCount=0,count=0,lastUpdated=<null
>,lastUpdatedTimestamp=<null>],byInstanceType={}],
repository=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@4b14e1ad[overall=
com.atlassian.jira.plugin.devstatus.summary.beans.CommitOverallBean@7a93ce17[count=0,
lastUpdated=<null>,lastUpdatedTimestamp=<null>],byInstanceType={}],
branch=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@17ac2419[overall=com.
atlassian.jira.plugin.devstatus.summary.beans.BranchOverallBean@1d993fa2[count=0,last
Updated=<null>,lastUpdatedTimestamp=<null
>],byInstanceType={}]},errors=[],configErrors=[]], devSummaryJson={"cachedValue":
{"errors":[],"configErrors":[],"summary":{"pullrequest":{"overall":{"count":
0,"lastUpdated":null,"stateCount":0,"state":"OPEN","details":{"openCount":
0,"mergedCount":0,"declinedCount":0,"total":0},"open":true},"byInstanceType":
{}},"build":{"overall":{"count":0,"lastUpdated":null,"failedBuildCount":
0,"successfulBuildCount":0,"unknownBuildCount":0},"byInstanceType":{}},"review":
{"overall":{"count":0,"lastUpdated":null,"stateCount":
0,"state":null,"dueDate":null,"overDue":false,"completed":false},"byInstanceType":
{}},"deployment-environment":{"overall":{"count":
0,"lastUpdated":null,"topEnvironments":[],"showProjects":false,"successfulCount":
0},"byInstanceType":{}},"repository":{"overall":{"count":
0,"lastUpdated":null},"byInstanceType":{}},"branch":{"overall":{"count":
0,"lastUpdated":null},"byInstanceType":{}}}},"isStale":true}}</customfield_11500>
      <timetracking/>
      <customfield_11104/>
      <customfield_11612/>
      <customfield_11733/>
      <customfield_11611/>
      <customfield_11732/>
      <customfield_11735/>
      <customfield_10800>0|i0bpfr:</customfield_10800>
      <customfield_11613/>
      <customfield_11734/>
```

```
      <attachment/>
      <customfield_10801/>
      <customfield_10802/>
      <customfield_11736/>
      <customfield_11739/>
      <summary>Demo: This is a low lvl api test</summary>
      <customfield_10000/>
      <customfield_10002/>
      <customfield_10003/>
      <customfield_11610/>
      <customfield_11731/>
      <customfield_10004>9223372036854775807</customfield_10004>
      <customfield_10400/>
      <customfield_11730/>
      <customfield_12415/>
      <customfield_11601/>
      <customfield_11722/>
      <customfield_12414/>
      <customfield_11600/>
      <customfield_11721/>
      <environment/>
      <customfield_12417/>
      <customfield_11603/>
      <customfield_12416/>
      <customfield_11602/>
      <customfield_11723/>
      <customfield_11605/>
      <customfield_12418/>
      <customfield_11604/>
      <duedate/>
      <customfield_11607/>
      <customfield_11606/>
      <customfield_11729/>
      <comment>
         <comments/>
         <maxResults>0</maxResults>
         <total>0</total>
         <startAt>0</startAt>
      </comment>
   </fields>
</JSON>
```

10.24.1.2 Get all Issues

Input

- Für diese Operation wird kein Input benötigt.
- Für diese Operation sind die Parameter `host`, `key` oder `query`, `username` und `password` oder `token` erforderlich.
- Der Projektschlüssel mit der Vorgangs-ID wird im Parameter `key` angegeben. Weitere Suchkriterien lassen sich im Parameter `query` definieren.

Output

- Wenn die Operation erfolgreich ausgeführt wurde, gibt der Adapter alle Vorgänge mit allen ausgefüllten Feldern aus.

Get all Issues

```
<JSON>
    <expand>schema,names</expand>
    <startAt>0</startAt>
    <maxResults>50</maxResults>
    <total>5</total>
    <issues>
        <e>
            <expand>operations,versionedRepresentations,editmeta,changelog,renderedFi
elds</expand>
            <id>82640</id>
            <self>https://jira.example.de/rest/api/2/issue/82640</self>
            <key>XVIERSC-89</key>
            <fields>
                <customfield_12130/>
                <customfield_12132/>
                <customfield_12131/>
                <fixVersions>
                    <e>
                        <self>https://jira.example.de/rest/api/2/version/15206</self>
                        <id>15206</id>
                        <description/>
                        <name>1.0.0</name>
                        <archived>false</archived>
                        <released>true</released>
                        <releaseDate>2020-11-27</releaseDate>
                    </e>
                </fixVersions>
                <customfield_12133/>
                <customfield_12411/>
                <resolution>
                    <self>https://jira.example.de/rest/api/2/resolution/1</self>
                    <id>1</id>
                    <description>A fix for this issue is checked into the tree and
tested.</description>
                    <name>Fixed</name>
                </resolution>
                <customfield_12410/>
                <customfield_12413/>
                <customfield_12412/>
                <customfield_12129/>
                <customfield_11711/>
                <customfield_12404/>
                <customfield_12128/>
                <customfield_11710/>
                <customfield_12403/>
                <customfield_11713/>
                <customfield_12406/>
                <customfield_11712/>
                <customfield_12405/>
                <customfield_11715/>
                <customfield_11714/>
```

```
                <customfield_12407/>
                <customfield_11719/>
                <lastViewed/>
                <customfield_12000/>
                <customfield_12121/>
                <customfield_12120/>
                <customfield_12123/>
                <customfield_12002/>
                <customfield_12001/>
                <customfield_12122/>
                <priority>
                    <self>https://jira.example.de/rest/api/2/priority/3</self>
                    <iconUrl>https://jira.example.de/images/icons/priorities/
major.svg</iconUrl>
                    <name>Major</name>
                    <id>3</id>
                </priority>
                <customfield_12400/>
                <customfield_12124/>
                <customfield_12127/>
                <labels>
                    <e>Blocker</e>
                </labels>
                <customfield_12401/>
                <customfield_11700/>
                <customfield_12118/>
                <customfield_12514/>
                <customfield_12513/>
                <customfield_11702/>
                <customfield_11701/>
                <customfield_12119/>
                <customfield_11704/>
                <aggregatetimeoriginalestimate/>
                <timeestimate/>
                <versions>
                    <e>
                        <self>https://jira.example.de/rest/api/2/version/15206</self>
                        <id>15206</id>
                        <description/>
                        <name>1.0.0</name>
                        <archived>false</archived>
                        <released>true</released>
                        <releaseDate>2020-11-27</releaseDate>
                    </e>
                </versions>
                <customfield_11703/>
                <customfield_11706/>
                <customfield_11705/>
                <customfield_11708/>
                <customfield_11707/>
                <issuelinks/>
                <customfield_11709/>
                <assignee>
                    <self>https://jira.example.de/rest/api/2/user?username=john.doe</
self>
```

```
                    <name>john.doe</name>
                    <key>john.doe</key>
                    <emailAddress>john.doe@example.de</emailAddress>
                    <avatarUrls>
                        <_48x48>https://jira.example.de/secure/useravatar?
ownerId=john.doe&avatarId=14916</_48x48>
                        <_24x24>https://jira.example.de/secure/useravatar?
size=small&ownerId=john.doe&avatarId=14916</_24x24>
                        <_16x16>https://jira.example.de/secure/useravatar?
size=xsmall&ownerId=john.doe&avatarId=14916</_16x16>
                        <_32x32>https://jira.example.de/secure/useravatar?
size=medium&ownerId=john.doe&avatarId=14916</_32x32>
                    </avatarUrls>
                    <displayName>John Doe</displayName>
                    <active>true</active>
                    <timeZone>Europe/Berlin</timeZone>
                </assignee>
                <status>
                    <self>https://jira.example.de/rest/api/2/status/10212</self>
                    <description/>
                    <iconUrl>https://jira.example.de/images/icons/statuses/
generic.png</iconUrl>
                    <name>Approved</name>
                    <id>10212</id>
                    <statusCategory>
                        <self>https://jira.example.de/rest/api/2/statuscategory/3</
self>
                        <id>3</id>
                        <key>done</key>
                        <colorName>green</colorName>
                        <name>Done</name>
                    </statusCategory>
                </status>
                <components>
                    <e>
                        <self>https://jira.example.de/rest/api/2/component/20010</
self>
                        <id>20010</id>
                        <name>Support Center</name>
                    </e>
                </components>
                <customfield_12110/>
                <customfield_12112/>
                <customfield_12111/>
                <customfield_12114/>
                <customfield_12510>
                    <self>https://jira.example.de/rest/api/2/customFieldOption/
11918</self>
                    <value>Ja</value>
                    <id>11918</id>
                </customfield_12510>
                <customfield_11300/>
                <customfield_12113/>
                <customfield_12116/>
                <customfield_12512/>
```

```
                <customfield_12115/>
                <customfield_12511/>
                <customfield_11810/>
                <customfield_12503/>
                <customfield_12502/>
                <customfield_11812/>
                <customfield_12505/>
                <customfield_11811/>
                <customfield_12507/>
                <aggregatetimeestimate/>
                <customfield_12506/>
                <customfield_12508/>
                <creator>
                    <self>https://jira.example.de/rest/api/2/user?username=john.doe</
self>
                    <name>john.doe</name>
                    <key>john.doe</key>
                    <emailAddress>john.doe@example.de</emailAddress>
                    <avatarUrls>
                        <_48x48>https://jira.example.de/secure/useravatar?
ownerId=john.doe&avatarId=14916</_48x48>
                        <_24x24>https://jira.example.de/secure/useravatar?
size=small&ownerId=john.doe&avatarId=14916</_24x24>
                        <_16x16>https://jira.example.de/secure/useravatar?
size=xsmall&ownerId=john.doe&avatarId=14916</_16x16>
                        <_32x32>https://jira.example.de/secure/useravatar?
size=medium&ownerId=john.doe&avatarId=14916</_32x32>
                    </avatarUrls>
                    <displayName>John Doe</displayName>
                    <active>true</active>
                    <timeZone>Europe/Berlin</timeZone>
                </creator>
                <subtasks/>
                <reporter>
                    <self>https://jira.example.de/rest/api/2/user?username=john.doe</
self>
                    <name>john.doe</name>
                    <key>john.doe</key>
                    <emailAddress>john.doe@example.de</emailAddress>
                    <avatarUrls>
                        <_48x48>https://jira.example.de/secure/useravatar?
ownerId=john.doe&avatarId=14916</_48x48>
                        <_24x24>https://jira.example.de/secure/useravatar?
size=small&ownerId=john.doe&avatarId=14916</_24x24>
                        <_16x16>https://jira.example.de/secure/useravatar?
size=xsmall&ownerId=john.doe&avatarId=14916</_16x16>
                        <_32x32>https://jira.example.de/secure/useravatar?
size=medium&ownerId=john.doe&avatarId=14916</_32x32>
                    </avatarUrls>
                    <displayName>John Doe</displayName>
                    <active>true</active>
                    <timeZone>Europe/Berlin</timeZone>
                </reporter>
                <aggregateprogress>
                    <progress>18900</progress>
```

```
                    <total>18900</total>
                    <percent>100</percent>
                </aggregateprogress>
                <customfield_12500/>
                <customfield_12612/>
                <customfield_12219/>
                <customfield_11801/>
                <customfield_11800/>
                <customfield_11803/>
                <customfield_11802/>
                <customfield_11805/>
                <customfield_11804/>
                <customfield_11807/>
                <customfield_11806/>
                <progress>
                    <progress>18900</progress>
                    <total>18900</total>
                    <percent>100</percent>
                </progress>
                <customfield_11809/>
                <votes>
                    <self>https://jira.example.de/rest/api/2/issue/XVIERSC-89/votes</
self>
                    <votes>0</votes>
                    <hasVoted>false</hasVoted>
                </votes>
                <customfield_11808/>
                <issuetype>
                    <self>https://jira.example.de/rest/api/2/issuetype/1</self>
                    <id>1</id>
                    <description>A problem which impairs or prevents the functions of
the product.</description>
                    <iconUrl>https://jira.example.de/secure/viewavatar?
size=xsmall&avatarId=11803&avatarType=issuetype</iconUrl>
                    <name>Bug</name>
                    <subtask>false</subtask>
                    <avatarId>11803</avatarId>
                </issuetype>
                <timespent>18900</timespent>
                <project>
                    <self>https://jira.example.de/rest/api/2/project/14505</self>
                    <id>14505</id>
                    <key>XVIERSC</key>
                    <name>X4 Solution Center</name>
                    <projectTypeKey>software</projectTypeKey>
                    <avatarUrls>
                        <_48x48>https://jira.example.de/secure/projectavatar?
pid=14505&avatarId=14707</_48x48>
                        <_24x24>https://jira.example.de/secure/projectavatar?
size=small&pid=14505&avatarId=14707</_24x24>
                        <_16x16>https://jira.example.de/secure/projectavatar?
size=xsmall&pid=14505&avatarId=14707</_16x16>
                        <_32x32>https://jira.example.de/secure/projectavatar?
size=medium&pid=14505&avatarId=14707</_32x32>
                    </avatarUrls>
```

```
                    <projectCategory>
                        <self>https://jira.example.de/rest/api/2/projectCategory/
10000</self>
                        <id>10000</id>
                        <description>Sample Products</description>
                        <name>Product</name>
                    </projectCategory>
                </project>
                <customfield_11001>
                    <self>https://jira.example.de/rest/api/2/customFieldOption/
10303</self>
                    <value>-1 - Unbekannt / Nicht bewertet</value>
                    <id>10303</id>
                </customfield_11001>
                <customfield_11002>999.0</customfield_11002>
                <aggregatetimespent>18900</aggregatetimespent>
                <customfield_12611/>
                <customfield_11400/>
                <customfield_12610/>
                <customfield_12601/>
                <customfield_10700/>
                <customfield_12604/>
                <customfield_12603/>
                <customfield_12606/>
                <customfield_12605/>
                <customfield_12608/>
                <resolutiondate>2020-10-22T11:12:46.000+0200</resolutiondate>
                <customfield_12607/>
                <customfield_12609/>
                <workratio>-1</workratio>
                <watches>
                    <self>https://jira.example.de/rest/api/2/issue/XVIERSC-89/
watchers</self>
                    <watchCount>2</watchCount>
                    <isWatching>false</isWatching>
                </watches>
                <created>2020-10-14T12:09:09.000+0200</created>
                <customfield_12200>
                    <self>https://jira.example.de/rest/api/2/customFieldOption/
11403</self>
                    <value>Störung</value>
                    <id>11403</id>
                </customfield_12200>
                <customfield_12600/>
                <customfield_10139>
                    <e>com.atlassian.greenhopper.service.sprint.Sprint@59bd9ff9[id=34
2,rapidViewId=121,state=CLOSED,name=Sprint 84
DE,startDate=2020-10-12T12:00:31.969+02:00,endDate=2020-10-23T13:00:00.000+02:00,comp
leteDate=2020-10-26T08:46:50.053+01:00,activatedDate=2020-10-12T12:00:31.969+02:00,se
quence=342,goal=Finish cutting of the videos
Update tutorials/demos on X4 Solution Center
Finish build automation for X4 Suite]</e>
                </customfield_10139>
                <updated>2020-10-22T11:12:46.000+0200</updated>
                <timeoriginalestimate/>
```

```
                <description>*enviroment*
 * test system ([http://10.128.22.99:8080/X4/webapp/Registered/login)]
 * version: pre 1.0.0 Build from 2020-10-13T09:46:34Z, revision 13d50b9a
 (according to: [http://10.128.22.99:8080/X4/webapp/SolutionCenter/Entdecken/
Discover)]

*expectation:* Mails can show special characters like "üÜ äÄ öÖ".

*current situation:* Mails can't show special characters like "üÜ öÖ äÄ".

*mail header snippet:*
{noformat}Subject: IN DEVELOPMENT:Ihr Ticket SUPCENTEST-11 wurde im Status Offen
 gespeichert.
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: quoted-printable {noformat}
*mail text (screenshot):*

  !image-2020-10-14-12-08-51-453.png!</description>
                <customfield_11100>
                    <self>https://jira.example.de/rest/api/2/customFieldOption/
10400</self>
                    <value>Product</value>
                    <id>10400</id>
                </customfield_11100>
                <customfield_11101>
                    <self>https://jira.example.de/rest/api/2/customFieldOption/
10404</self>
                    <value>-1 - Unbekannt / Nicht bewertet</value>
                    <id>10404</id>
                </customfield_11101>
                <customfield_11740/>
```

```
                <customfield_11500>{summaryBean=com.atlassian.jira.plugin.devstatus.r
est.SummaryBean@31b2d52c[summary={pullrequest=com.atlassian.jira.plugin.devstatus.res
t.SummaryItemBean@24aa33fa[overall=PullRequestOverallBean{stateCount=2,
state='MERGED', details=PullRequestOverallDetails{openCount=0, mergedCount=2,
declinedCount=0}},byInstanceType={stash=com.atlassian.jira.plugin.devstatus.summary.b
eans.ObjectByInstanceTypeBean@7e649bc6[count=2,name=Bitbucket Server]}],
build=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@66337825[overall=com.a
tlassian.jira.plugin.devstatus.summary.beans.BuildOverallBean@228f2b11[failedBuildCou
nt=0,successfulBuildCount=0,unknownBuildCount=0,count=0,lastUpdated=<null
>,lastUpdatedTimestamp=<null>],byInstanceType={}],
review=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@6eaec696[overall=com.
atlassian.jira.plugin.devstatus.summary.beans.ReviewsOverallBean@6031763d[stateCount=
0,state=<null>,dueDate=<null>,overDue=false,count=0,lastUpdated=<null
>,lastUpdatedTimestamp=<null>],byInstanceType={}], deployment-
environment=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@1dbed84e[overall
=com.atlassian.jira.plugin.devstatus.summary.beans.DeploymentOverallBean@4632af4b[top
Environments=[],showProjects=false,successfulCount=0,count=0,lastUpdated=<null
>,lastUpdatedTimestamp=<null>],byInstanceType={}],
repository=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@5eb04b6d[overall=
com.atlassian.jira.plugin.devstatus.summary.beans.CommitOverallBean@35db5e18[count=5,
lastUpdated=2020-10-27T11:40:21.000+0100,lastUpdatedTimestamp=2020-10-27T11:40:21.000
+01:00],byInstanceType={stash=com.atlassian.jira.plugin.devstatus.summary.beans.Objec
tByInstanceTypeBean@357dd6cf[count=5,name=Bitbucket Server]}],
branch=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@dd72830[overall=com.a
tlassian.jira.plugin.devstatus.summary.beans.BranchOverallBean@51e0f1c4[count=0,lastU
pdated=<null>,lastUpdatedTimestamp=<null
>],byInstanceType={}]},errors=[],configErrors=[]], devSummaryJson={"cachedValue":
{"errors":[],"configErrors":[],"summary":{"pullrequest":{"overall":{"count":
2,"lastUpdated":"2020-10-19T16:42:21.304+0200","stateCount":
2,"state":"MERGED","details":{"openCount":0,"mergedCount":2,"declinedCount":
0,"total":2},"open":false},"byInstanceType":{"stash":{"count":2,"name":"Bitbucket
Server"}}},"build":{"overall":{"count":0,"lastUpdated":null,"failedBuildCount":
0,"successfulBuildCount":0,"unknownBuildCount":0},"byInstanceType":{}},"review":
{"overall":{"count":0,"lastUpdated":null,"stateCount":
0,"state":null,"dueDate":null,"overDue":false,"completed":false},"byInstanceType":
{}},"deployment-environment":{"overall":{"count":
0,"lastUpdated":null,"topEnvironments":[],"showProjects":false,"successfulCount":
0},"byInstanceType":{}},"repository":{"overall":{"count":
5,"lastUpdated":"2020-10-27T11:40:21.000+0100"},"byInstanceType":{"stash":{"count":
5,"name":"Bitbucket Server"}}},"branch":{"overall":{"count":
0,"lastUpdated":null},"byInstanceType":{}}}},"isStale":true}}</customfield_11500>
                <customfield_11104/>
                <customfield_11612/>
                <customfield_11733/>
                <customfield_11611/>
                <customfield_11732/>
                <customfield_11735/>
                <customfield_10800>0|i0avz3:</customfield_10800>
                <customfield_11613/>
                <customfield_11734/>
                <customfield_10801/>
                <customfield_10802/>
                <customfield_11736/>
                <customfield_11739/>
```

```
                <summary>Support Center - Mails can't show german special characters
like "üÜ öÖ äÄ"</summary>
                <customfield_10000/>
                <customfield_10002/>
                <customfield_11610/>
                <customfield_11731/>
                <customfield_10400/>
                <customfield_10004>9223372036854775807</customfield_10004>
                <customfield_11730/>
                <customfield_12415/>
                <customfield_11601/>
                <customfield_11722/>
                <customfield_12414/>
                <customfield_11600/>
                <customfield_11721/>
                <environment/>
                <customfield_12417/>
                <customfield_11603/>
                <customfield_12416/>
                <customfield_11602/>
                <customfield_11723/>
                <customfield_11605/>
                <customfield_12418/>
                <customfield_11604/>
                <duedate/>
                <customfield_11607/>
                <customfield_11606/>
                <customfield_11729/>
            </fields>
        </e>

        <!-- more issues -->

    </issues>
</JSON>
```

10.24.1.3 Create Issues

Input

- Für diese Operation sind die Parameter `host`, `key`, `username` und `password` oder `token` erforderlich.

Create Issue

```
<RootElement xmlns:j="http://www.softproject.de/schemas/2020/json-converter">
    <fields>
        <project>
            <key>XVIERSC</key>
        </project>
        <summary>Issue title</summary>
        <description>Issue description</description>
        <components j:mode="array">
            <Object>
                <name>Demos</name>
            </Object>
        </components>
        <issuetype>
            <name>Story</name>
        </issuetype>
    </fields>
</RootElement>
```

Output

Create Issue

```
<JSON>
   <id>87606</id>
   <key>XVIERSC-174</key>
   <self>https://jira.softproject.de/rest/api/2/issue/87606</self>
</JSON>
```

10.24.1.4 Edit Issues

Input

- Für diese Operation sind die Parameter `host`, `key`, `username` und `password` oder `token` erforderlich.

Edit Issue

```
<RootElement xmlns:j="http://www.softproject.de/schemas/2020/json-converter">
    <fields>
        <customfield_11002>987</customfield_11002>
    </fields>
</RootElement>
```

Output

Edit Issue

```
<Message>The operation was successful!</Message>
```

10.24.1.5 Add Attachment

Input

- Für diese Operation sind die Parameter `host`, `key`, `username` und `password` oder `token` erforderlich.
- Die Datei, die angehängt werden soll, muss Base64-kodiert sein.
- Es können auch mehrere Dateien gleichzeitig angehängt werden.

Add Attachment

```
<?xml version="1.0" encoding="UTF-8"?>
<RootElement>
    <attachments>
        <attachment>
            <fileName>test.txt</fileName>
            <fileData>SGFsbG8gV2VsdCEhIQ==</fileData>
        </attachment>
        <attachment>
            <fileName>test.xml</fileName>

<fileData>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxyb290Pg0KPHRlc3Q+SGFsbG8gV2VsdCE8L3Rlc3Q+DQo8L3Jvb3Q+</fileData>
        </attachment>
    </attachments>
</RootElement>
```

Output

- Wenn die Operation erfolgreich ausgeführt wurde, dann gibt der Adapter die globale Jira-ID des Anhangs aus.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<JSON>
    <e>
        <self>https://jira.example.de/rest/api/2/attachment/57798</self>
        <id>57798</id>
        <filename>test.txt</filename>
        <author>
            <self>https://jira.example.de/rest/api/2/user?
username=x4solution.center</self>
            <name>x4solution.center</name>
            <key>JIRAUSER12810</key>
            <emailAddress>x4solution.center@example.de</emailAddress>
            <avatarUrls>
                <_48x48>https://jira.example.de/secure/useravatar?
ownerId=JIRAUSER12810&avatarId=14718</_48x48>
                <_24x24>https://jira.example.de/secure/useravatar?
size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
                <_16x16>https://jira.example.de/secure/useravatar?
size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
                <_32x32>https://jira.example.de/secure/useravatar?
size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
            </avatarUrls>
            <displayName>X4 Solution Center</displayName>
            <active>true</active>
            <timeZone>Europe/Berlin</timeZone>
        </author>
        <created>2021-02-05T14:48:09.864+0100</created>
        <size>13</size>
        <content>https://jira.example.de/secure/attachment/57798/
test.txt</content>
    </e>
    <e>
        <self>https://jira.example.de/rest/api/2/attachment/57799</self>
        <id>57799</id>
        <filename>test.xml</filename>
        <author>
            <self>https://jira.example.de/rest/api/2/user?
username=x4solution.center</self>
            <name>x4solution.center</name>
            <key>JIRAUSER12810</key>
            <emailAddress>x4solution.center@example.de</emailAddress>
            <avatarUrls>
                <_48x48>https://jira.example.de/secure/useravatar?
ownerId=JIRAUSER12810&avatarId=14718</_48x48>
                <_24x24>https://jira.example.de/secure/useravatar?
size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
                <_16x16>https://jira.example.de/secure/useravatar?
size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
                <_32x32>https://jira.example.de/secure/useravatar?
size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
            </avatarUrls>
            <displayName>X4 Solution Center</displayName>
            <active>true</active>
            <timeZone>Europe/Berlin</timeZone>
```

```
44          </author>
45          <created>2021-02-05T14:48:09.867+0100</created>
46          <size>81</size>
47          <content>https://jira.example.de/secure/attachment/57799/
test.xml</content>
48      </e>
49  </JSON>
```

10.24.1.6 Remove Attachment

Input

- Für diese Operation wird kein Input benötigt.
- Für diese Operation sind die Parameter `host`, `username`, `password` oder `token` und `id` erforderlich.
- Die ID des Anhangs wird im Parameter `id` angegeben.

Output

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Message>The operation was successful!</Message>
```

10.24.1.7 Get all Comments

Input

- Für diese Operation wird kein Input benötigt.
- Für diese Operation sind die Parameter `host`, `key`, `username` und `password` oder `token` erforderlich.
- Der Projektschlüssel mit der Vorgangs-ID wird im Parameter `key` angegeben.

Output

```
<?xml version="1.0" encoding="UTF-8"?>
<JSON>
    <startAt>0</startAt>
    <maxResults>50</maxResults>
    <total>2</total>
    <comments>
        <e>
            <self>https://jira.example.de/rest/api/2/issue/87606/comment/137249</self>
            <id>137249</id>
            <author>
                <self>https://jira.example.de/rest/api/2/user?username=x4solution.center</self>
                <name>x4solution.center</name>
                <key>JIRAUSER12810</key>
                <emailAddress>x4solution.center@example.de</emailAddress>
                <avatarUrls>
                    <_48x48>https://jira.example.de/secure/useravatar?ownerId=JIRAUSER12810&avatarId=14718</_48x48>
                    <_24x24>https://jira.example.de/secure/useravatar?size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
                    <_16x16>https://jira.example.de/secure/useravatar?size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
                    <_32x32>https://jira.example.de/secure/useravatar?size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
                </avatarUrls>
                <displayName>X4 Solution Center</displayName>
                <active>true</active>
                <timeZone>Europe/Berlin</timeZone>
            </author>
            <body>This is an edited comment for an issue</body>
            <updateAuthor>
                <self>https://jira.example.de/rest/api/2/user?username=x4solution.center</self>
                <name>x4solution.center</name>
                <key>JIRAUSER12810</key>
                <emailAddress>x4solution.center@example.de</emailAddress>
                <avatarUrls>
                    <_48x48>https://jira.example.de/secure/useravatar?ownerId=JIRAUSER12810&avatarId=14718</_48x48>
                    <_24x24>https://jira.example.de/secure/useravatar?size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
                    <_16x16>https://jira.example.de/secure/useravatar?size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
                    <_32x32>https://jira.example.de/secure/useravatar?size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
                </avatarUrls>
                <displayName>X4 Solution Center</displayName>
                <active>true</active>
                <timeZone>Europe/Berlin</timeZone>
            </updateAuthor>
            <created>2021-02-05T14:04:21.896+0100</created>
            <updated>2021-02-05T14:42:33.967+0100</updated>
        </e>
```

```
        <e>
            <self>https://jira.example.de/rest/api/2/issue/87606/comment/
137255</self>
            <id>137255</id>
            <author>
                <self>https://jira.example.de/rest/api/2/user?
username=x4solution.center</self>
                <name>x4solution.center</name>
                <key>JIRAUSER12810</key>
                <emailAddress>x4solution.center@example.de</emailAddress>
                <avatarUrls>
                    <_48x48>https://jira.example.de/secure/useravatar?
ownerId=JIRAUSER12810&avatarId=14718</_48x48>
                    <_24x24>https://jira.example.de/secure/useravatar?
size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
                    <_16x16>https://jira.example.de/secure/useravatar?
size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
                    <_32x32>https://jira.example.de/secure/useravatar?
size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
                </avatarUrls>
                <displayName>X4 Solution Center</displayName>
                <active>true</active>
                <timeZone>Europe/Berlin</timeZone>
            </author>
            <body>This is a second comment for an issue</body>
            <updateAuthor>
                <self>https://jira.example.de/rest/api/2/user?
username=x4solution.center</self>
                <name>x4solution.center</name>
                <key>JIRAUSER12810</key>
                <emailAddress>x4solution.center@example.de</emailAddress>
                <avatarUrls>
                    <_48x48>https://jira.example.de/secure/useravatar?
ownerId=JIRAUSER12810&avatarId=14718</_48x48>
                    <_24x24>https://jira.example.de/secure/useravatar?
size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
                    <_16x16>https://jira.example.de/secure/useravatar?
size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
                    <_32x32>https://jira.example.de/secure/useravatar?
size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
                </avatarUrls>
                <displayName>X4 Solution Center</displayName>
                <active>true</active>
                <timeZone>Europe/Berlin</timeZone>
            </updateAuthor>
            <created>2021-02-05T14:43:37.639+0100</created>
            <updated>2021-02-05T14:43:37.639+0100</updated>
        </e>
    </comments>
</JSON>
```

10.24.1.8 Get Comment

Input

- Für diese Operation wird kein Input benötigt.
- Für diese Operation sind die Parameter `host`, `key`, `username`, `password` oder `token` und `id` erfo rderlich.
- Der Projektschlüssel mit der Vorgangs-ID wird im Parameter `key` angegeben, die entsprechende ID des Kommentars im Parameter `id` .

Output

```xml
<?xml version="1.0" encoding="UTF-8"?>
<JSON>
    <self>https://jira.example.de/rest/api/2/issue/87606/comment/137249</self>
    <id>137249</id>
    <author>
        <self>https://jira.example.de/rest/api/2/user?username=x4solution.center</self>
        <name>x4solution.center</name>
        <key>JIRAUSER12810</key>
        <emailAddress>x4solution.center@example.de</emailAddress>
        <avatarUrls>
            <_48x48>https://jira.example.de/secure/useravatar?ownerId=JIRAUSER12810&avatarId=14718</_48x48>
            <_24x24>https://jira.example.de/secure/useravatar?size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
            <_16x16>https://jira.example.de/secure/useravatar?size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
            <_32x32>https://jira.example.de/secure/useravatar?size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
        </avatarUrls>
        <displayName>X4 Solution Center</displayName>
        <active>true</active>
        <timeZone>Europe/Berlin</timeZone>
    </author>
    <body>This is an edited comment for an issue</body>
    <updateAuthor>
        <self>https://jira.example.de/rest/api/2/user?username=x4solution.center</self>
        <name>x4solution.center</name>
        <key>JIRAUSER12810</key>
        <emailAddress>x4solution.center@example.de</emailAddress>
        <avatarUrls>
            <_48x48>https://jira.example.de/secure/useravatar?ownerId=JIRAUSER12810&avatarId=14718</_48x48>
            <_24x24>https://jira.example.de/secure/useravatar?size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
            <_16x16>https://jira.example.de/secure/useravatar?size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
            <_32x32>https://jira.example.de/secure/useravatar?size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
        </avatarUrls>
        <displayName>X4 Solution Center</displayName>
        <active>true</active>
        <timeZone>Europe/Berlin</timeZone>
    </updateAuthor>
    <created>2021-02-05T14:04:21.896+0100</created>
    <updated>2021-02-05T14:42:33.967+0100</updated>
</JSON>
```

10.24.1.9 Add Comment

Input

- Für diese Operation sind die Parameter `host`, `key`, `username` und `password` oder `token` erforderlich.

```
<RootElement xmlns:j="http://www.softproject.de/schemas/2020/json-converter">
    <body>This is a comment for an issue</body>
</RootElement>
```

Output

```
<JSON>
    <self>https://jira.example.de/rest/api/2/issue/87606/comment/137249</self>
    <id>137249</id>
    <author>
        <self>https://jira.example.de/rest/api/2/user?username=x4solution.center</
self>
        <name>x4solution.center</name>
        <key>JIRAUSER12810</key>
        <emailAddress>x4solution.center@example.de</emailAddress>
        <avatarUrls>
            <_48x48>https://jira.example.de/secure/useravatar?
ownerId=JIRAUSER12810&avatarId=14718</_48x48>
            <_24x24>https://jira.example.de/secure/useravatar?
size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
            <_16x16>https://jira.example.de/secure/useravatar?
size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
            <_32x32>https://jira.example.de/secure/useravatar?
size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
        </avatarUrls>
        <displayName>X4 Solution Center</displayName>
        <active>true</active>
        <timeZone>Europe/Berlin</timeZone>
    </author>
    <body>This is a comment for an issue</body>
    <updateAuthor>
        <self>https://jira.example.de/rest/api/2/user?username=x4solution.center</
self>
        <name>x4solution.center</name>
        <key>JIRAUSER12810</key>
        <emailAddress>x4solution.center@example.de</emailAddress>
        <avatarUrls>
            <_48x48>https://jira.example.de/secure/useravatar?
ownerId=JIRAUSER12810&avatarId=14718</_48x48>
            <_24x24>https://jira.example.de/secure/useravatar?
size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
            <_16x16>https://jira.example.de/secure/useravatar?
size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
            <_32x32>https://jira.example.de/secure/useravatar?
size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
        </avatarUrls>
        <displayName>X4 Solution Center</displayName>
        <active>true</active>
        <timeZone>Europe/Berlin</timeZone>
    </updateAuthor>
    <created>2021-02-05T14:04:21.896+0100</created>
    <updated>2021-02-05T14:04:21.896+0100</updated>
</JSON>
```

10.24.1.10 Edit Comment

Input

- Für diese Operation sind die Parameter `host`, `key`, `username`, `password` oder `token` und `id` erforderlich.
- Der Projektschlüssel mit der Vorgangs-ID wird im Parameter `key` angegeben, die entsprechende ID des Kommentars im Parameter `id`.

```
<?xml version="1.0" encoding="UTF-8"?>
<RootElement xmlns:j="http://www.example.de/schemas/2020/json-converter">
    <body>This is an edited comment for an issue</body>
</RootElement>
```

Output

```
<?xml version="1.0" encoding="UTF-8"?>
<JSON>
    <self>https://jira.example.de/rest/api/2/issue/87606/comment/137249</
self>
    <id>137249</id>
    <author>
        <self>https://jira.example.de/rest/api/2/user?
username=x4solution.center</self>
        <name>x4solution.center</name>
        <key>JIRAUSER12810</key>
        <emailAddress>x4solution.center@example.de</emailAddress>
        <avatarUrls>
            <_48x48>https://jira.example.de/secure/useravatar?
ownerId=JIRAUSER12810&avatarId=14718</_48x48>
            <_24x24>https://jira.example.de/secure/useravatar?
size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
            <_16x16>https://jira.example.de/secure/useravatar?
size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
            <_32x32>https://jira.example.de/secure/useravatar?
size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
        </avatarUrls>
        <displayName>X4 Solution Center</displayName>
        <active>true</active>
        <timeZone>Europe/Berlin</timeZone>
    </author>
    <body>This is an edited comment for an issue</body>
    <updateAuthor>
        <self>https://jira.example.de/rest/api/2/user?
username=x4solution.center</self>
        <name>x4solution.center</name>
        <key>JIRAUSER12810</key>
        <emailAddress>x4solution.center@example.de</emailAddress>
        <avatarUrls>
            <_48x48>https://jira.example.de/secure/useravatar?
ownerId=JIRAUSER12810&avatarId=14718</_48x48>
            <_24x24>https://jira.example.de/secure/useravatar?
size=small&ownerId=JIRAUSER12810&avatarId=14718</_24x24>
            <_16x16>https://jira.example.de/secure/useravatar?
size=xsmall&ownerId=JIRAUSER12810&avatarId=14718</_16x16>
            <_32x32>https://jira.example.de/secure/useravatar?
size=medium&ownerId=JIRAUSER12810&avatarId=14718</_32x32>
        </avatarUrls>
        <displayName>X4 Solution Center</displayName>
        <active>true</active>
        <timeZone>Europe/Berlin</timeZone>
    </updateAuthor>
    <created>2021-02-05T14:04:21.896+0100</created>
    <updated>2021-02-05T14:39:58.930+0100</updated>
</JSON>
```

10.24.1.11 Search

Input

- Für diese Operation wird kein Input benötigt.
- Für diese Operation sind die Parameter `host`, `username`, `password` oder `token` und `query` erforderlich.
- Weitere Suchkriterien lassen sich im Parameter `query` definieren.

Output

```
<?xml version="1.0" encoding="UTF-8"?>
<IssueCollection count="50" offset="0" total="143">
    <Issue href="https://jira.example.de/rest/api/latest/issue/51941" id="51941"
           key="EXAMPLESC-149">
        <FixVersions/>
        <Labels/>
        <Remaining>0.25</Remaining>
        <AffectedVersions/>
        <Status href="https://jira.example.de/rest/api/2/status/10206" id="10206">New</Status>
        <Creator href="https://jira.example.de/rest/api/2/user?username=john.doe">
            <UserName>john.doe</UserName>
            <DisplayName>John Doe</DisplayName>
        </Creator>
        <Type href="https://jira.example.de/rest/api/2/issuetype/7" id="7">Story</Type>
        <Project href="https://jira.example.de/rest/api/2/project/14505" id="14505">Sample project</Project>
        <Rank>400.0</Rank>
        <CreatedOn>2018-08-30T12:54:42.000+02:00</CreatedOn>
        <ChangedOn>2020-12-15T09:08:54.000+01:00</ChangedOn>
        <Estimation>0.25</Estimation>
        <Description>
            <!--Issue description-->
        </Description>
        <Account href="" id=""/>
        <Title>Issue title</Title>
    </Issue>

    <!-- more issues -->

</IssueCollection>
```

10.24.1.12 Execute Transition

Input

- Für diese Operation sind die Parameter `host`, `key`, `username` und `password` oder `token` erforderlich.

Execute Transition

```
<RootElement xmlns:j="http://www.softproject.de/schemas/2020/json-converter">
    <transition>
        <!-- Set to in Progress -->
        <id j:mode="literal">11</id>
        <!-- Set to New -->
        <!-- <id j:mode="literal">51</id> -->
        <!-- Set to Implemented -->
        <!-- <id j:mode="literal">121</id> -->
    </transition>
</RootElement>
```

Output

Wurde die Operation erfolgreich ausgeführt, gibt der Adapter `OK` aus.

Execute Transition

```
<Ok>
    <Message>Executing transition on issue was successful</Message>
</Ok>
```

Andernfalls wird der Status `-1` ausgegeben. Wenn es möglich ist, mögliche Übergänge abzurufen, werden diese ausgegeben (bei falscher Übergangs-ID hilft das, die richtige ID zu finden).

Execute Transition

```
<Error>
   <Message>Error on Executing transition, maybe the called transition is not possible in the state of the issue, please check possible transitions</Message>
   <PossibleTransitions>
      <Object>
         <Value name="expand" type="String">transitions</Value>
         <Array name="transitions">
            <Object>
               <Value name="id" type="String">61</Value>
               <Value name="name" type="String">Zurück</Value>
               <Object name="to">...
```

10.24.1.13 Get Transition

Input

- Für diese Operation wird kein Input benötigt.
- Für diese Operation sind die Parameter `host`, `key`, `username` und `password` oder `token` erforderlich.
- Der Projektschlüssel mit der Vorgangs-ID wird im Parameter `key` angegeben.

Output

```
<?xml version="1.0" encoding="UTF-8"?>
<JSON>
    <expand>transitions</expand>
    <transitions>
        <e>
            <id>11</id>
            <name>Bearbeitung</name>
            <to>
                <self>https://jira.example.de/rest/api/2/status/3</self>
                <description>This issue is being actively worked on at the moment by the assignee.</description>
                <iconUrl>https://jira.example.de/images/icons/statuses/inprogress.png</iconUrl>
                <name>In Progress</name>
                <id>3</id>
                <statusCategory>
                    <self>https://jira.example.de/rest/api/2/statuscategory/4</self>
                    <id>4</id>
                    <key>indeterminate</key>
                    <colorName>yellow</colorName>
                    <name>In Progress</name>
                </statusCategory>
            </to>
        </e>
    </transitions>
</JSON>
```

10.24.2 Atlassian Jira Connector ReST API

`Atlassian Jira Connector`: Bietet einen Low-Level-Zugriff auf die Atlassian Jira ReST-API.
Properties (Eigenschaften)

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • `Execute`: Adapter mit der vorgenommenen Parametrierung ausführen

Parameter

`Adapter`	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.nativeadapter.NativeAdapter`* : Hauptklasse (Standard)
`Repository`	Virtuelles Repository des Adapter-Projekts. Der Wert kann nicht geändert werden.
`host`	*Erforderlich*. URL des Jira-Servers **Mögliche Werte:** Beliebige gültige URL, z. B. *`jira.example.com`*
`port`	Port der ReST-API **Mögliche Werte:** • Beliebige positive ganze Zahl • *443* : Standardport
`path`	*Erforderlich*. Ressourcen-Pfad für die Anfrage **Mögliche Werte:** URL-Part in Jira (Weitere Informationen unter https://docs.atlassian.com/software/jira/docs/api/REST/7.13.0/#api/2), z.B. */rest/api/2/{issueIdOrKey}*
`secure`	Angabe, ob die Kommunikation über HTTPS stattfinden soll oder nicht **Mögliche Werte:** `true` (Standard) / `false`
`username`	*Erforderlich*. Jira-Benutzer, der für die Authentifizierung verwendet wird
`password`	*Erforderlich, falls `token` nicht verwendet wird*. Passwort des Jira-Benutzers für die Authentifizierung
`token`	*Erforderlich, falls `password` nicht verwendet wird*. Für die Authentifizierung verwendetes Token. Das Token kann anstelle eines Passworts verwendet werden.

`method`	*Erforderlich*. HTTP-Methode, die für die Anfrage verwendet wird **Mögliche Werte:** • `Get` (Standard) • `Post` • `Put` • `Delete` • `Patch`
`query`	URL-kodierte Abfrage, z. B. *query="project={projectKey} ORDER BY createdDate ASC"*

Statuswerte

1	>Die Operation wurde erfolgreich ausgeführt.
-1	Es kam zu einem Fehler während der Ausführung.

Input

Der Adapter erwartet einen Input im XML-Format. Im Wurzelelement muss die Namespace-Definition `xmlns:j=http://www.softproject.de/schemas/2020/json-converter` enthalten sein. Der genaue Aufbau findet sich unter https://docs.atlassian.com/software/jira/docs/api/REST/7.13.0/#api/2.

Output

Der Adapter gibt ein Dokument mit der Struktur, die unter https://docs.atlassian.com/software/jira/docs/api/REST/7.13.0/#api/2 beschrieben wird, aber im XML-Format. Wenn kein Inhalt vorhanden ist, der ausgegeben werden kann, dann wird die Nachricht `The operation was successful!` ausgegeben.

10.24.3 Global Parameter Initializer (früher: Custom Placeholder Initializer) (veraltet)

`Global Parameter Initializer` (früher: `Custom Placeholder Initializer`): Verwaltet die globalen Parameter im aktuellen Prozess. Das Laden und Entladen von globalen Parameter aus dem Prozesskontext kann über Adapter-Parameter oder ein XML-Dokument gesteuert werden.

Eigenschaften (Properties)

`Operation`	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • `Load`: Lädt die im Parameter `groups` definierten globalen Parameter in den Prozesskontext (*Input-XML = Output-XML*) • `Unload`: Entfernt die im Parameter `groups` definierten globalen Parameter aus dem Prozesskontext (*Input-XML = Output-XML*) • `LoadFromXML`: Lädt die globalen Parameter aus dem XML-Dokument in den Prozesskontext • `UnloadFromXML`: Entfernt die globalen Parameter aus dem XML-Dokument aus dem Prozesskontext • `Export`: Gibt alle aktuell gesetzten globalen Parameter im Prozesskontext aus (benötigt kein Input-Dokument)

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.globalparameter.GlobalParameterInitializerAdapter*: Hauptklasse (Standard)
groups	Liste der zu ladenden globalen Parameter-Gruppen aus dem Parameter-Speicher (**Global Parameter Configuration**; früher: **Custom Placeholder Storage Editor**). Werte können mit Komma oder Semikolon getrennt werden. *Beispiel:* `DEFAULT, Prod, Dev, Test`

Statuswerte

1 (successful)	Die Operation war erfolgreich
0 (empty)	Es wurden keine globalen Parameter in den Prozesskontext geladen
-1 (failed)	Die Operation schlug aufgrund eines technischen Fehlers fehl

Input

Die Operationen `LoadFromXML` und `UnloadFromXML` erwarten folgendes XML-Format:

LoadFromXML

```
<?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEV">
        <Key name="message">Hallo Entwickler!</Key>
    </Group>
    <Group name="DEFAULT">
        <Key name="message">Hallo Welt!</Key>
    </Group>
</Placeholder>
```

Es werden die zwei globalen Parameter-Gruppen `DEFAULT` und `DEV` mit dem jeweiligen Parameter *`message`* im Parameterkontext abgelegt.

UnloadFromXML

```
<?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEV" />
    <Group name="DEFAULT">
        <Key name="message" />
    </Group>
</Placeholder>
```

Alle globalen Parameter der Gruppe `DEV` und der Parameter *`message`* aus der gloabeln Parameter-Gruppe *`DEFAULT`* werden aus dem Prozesskontext entfernt.

Ausgabe

Die Operation `Get` gibt alle aktuellen globalen Parameter im Prozesskontext aus. Die Operationen `LoadFromXML` und `UnloadFromXML` geben die verarbeiteten Elemente aus.

```
<?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEV">
        <Key name="message">Hallo Entwickler!</Key>
    </Group>
    <Group name="DEFAULT">
        <Key name="message">Hallo Welt!</Key>
    </Group>
</Placeholder>
```

10.24.4 MongoDB Connector

10.24.4.1 Über den Adapter

`MongoDB Connector`: Stellt eine Verbindung zu einer MongoDB her und führt ein oder mehrere Datenbank-Statements aus, die als Input-XML-Dokument mit adapterspezifischen Strukturen übergeben wurden.

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.mongodb.MongoAdapter*: Hauptklasse (Standard)
ConnectionName	Alias/Referenzname für die Verbindung. Entspricht dem `Connection name` aus der Verbindungskonfiguration. Weitere Informationen hierzu finden Sie unter Verbindung konfigurieren. **Mögliche Werte:** Name der Verbindung (z. B. *#default* (Standard))
ReadPreference	Read-Preference-Modus. Dieser Parameter definiert, wie Reads bearbeitet werden. **Mögliche Werte:** • *`PRIMARY`*: Alle Operationen lesen aus dem `primary`-Knoten des aktuellen Replica Sets. • *`PRIMARY_PREFERRED`*: In den meisten Fällen lesen alle Operationen aus dem `primary`-Knoten. Wenn dieser Knoten nicht verfügbar ist, lesen die Operationen aus den `secondary`-Knoten (Standard). • *`SECONDARY`*: Alle Operationen lesen aus den `secondary`-Knoten des aktuellen Replica Sets. • *`SECONDARY_PREFERRED`*: In den meisten Fällen lesen alle Operationen aus den `secondary`-Knoten. Wenn kein `secondary`-Knoten verfügbar ist, lesen die Operationen aus dem `primary`-Knoten. • *`NEAREST`*: Die Operationen lesen unabhängig vom Knotentyp aus dem Knoten des Replica Sets, der die niedrigste Netzwerk-Latenzzeit aufweist. ⓘ Zusätzliche Informationen zu den Read-Preference-Modi finden Sie unter: • http://docs.mongodb.org/manual/core/read-preference/#read-preference-modes • http://api.mongodb.org/java/current/com/mongodb/ReadPreference.html#method_summary

WriteConcern	Gibt an, ob eine `write`-Operation erfolgreich ausgeführt wurde. Mit Write Concerns können Eingabefehler oder ungültige `mongod`-Instanzen ermittelt werden. **Mögliche Werte:** • `ERRORS_IGNORED`: Es werden keine Exceptions ausgeworfen, auch nicht bei Netzwerkproblemen. ⓘ Dieser Write Concern ist veraltet und verfügt über keinen Ersatz. Am ehesten vergleichbar ist der Write Concern `UNACKNOWLEDGED`. Dabei müssen jedoch alle Exceptions des Typs *MongoSocketException* über *catch and ignore* abgefangen werden. • `ACKNOWLEDGED`: Die Operation wartet auf die Bestätigung des primären Servers, bevor sie ein Ergebnis ausgibt. Exceptions werden sowohl bei Netzwerkproblemen als auch bei Serverfehlern ausgeworfen. • `UNACKNOWLEDGED`: Die Operation gibt das Ergebnis aus, sobald die Nachricht in den Socket geschrieben wird. Exceptions werden lediglich bei Netzwerkproblemen, jedoch nicht bei Serverfehlern ausgeworfen. • `FSYNCED`: Die Operation wartet, bis der Server sichergestellt hat, dass die Daten auf der Festplatte angekommen sind. Exceptions werden sowohl bei Netzwerkproblemen als auch bei Serverfehlern ausgeworfen. • `JOURNALED`: Die Operation wartet, bis der Server die Daten zu den Journal-Files committet. Exceptions werden sowohl bei Netzwerkproblemen als auch bei Serverfehlern ausgeworfen. ⓘ **Beachten Sie:** • Das Journaling muss aktiviert sein, um diesen Write Concern zu verwenden. • Bei diesem Write Concern müssen die Operationen bis zum nächsten Journal-Commit warten. Um die Wartezeit für diese Operationen zu verkürzen, erhöht *MongoDB* die Commit-Häufigkeit. Weitere Informationen hierzu finden Sie unter commitIntervalMs. • `REPLICA_ACKNOWLEDGED`: Die Operation wartet auf mindestens zwei Server, bevor sie ausgeführt wird. Exceptions werden sowohl bei Netzwerkproblemen als auch bei Serverfehlern ausgeworfen. • `NONE`: Es werden keine Exceptions ausgeworfen, auch nicht bei Netzwerkproblemen.

> Dieser Write Concern wurde durch den Write Concern `ERRORS_IGNORED` und kann in kommenden Releases ggf. veraltet sein.

- `NORMAL`: Die Operation gibt das Ergebnis aus, sobald die Nachricht in den Socket geschrieben wird. Exceptions werden lediglich bei Netzwerkproblemen ausgeworfen, jedoch nicht bei Serverfehlern.

  > Dieser Write Concern wurde durch den Write Concern `UNACKNOWLEDGED` ersetzt und kann in kommenden Releases ggf. veraltet sein .

- `SAFE`: Die Operation wartet auf die Bestätigung des primären Servers, bevor sie ein Ergebnis ausgibt. Exceptions werden sowohl bei Netzwerkproblemen als auch bei Serverfehlern ausgeworfen.

  > Dieser Write Concern wurde durch den Write Concern `ACKNOWLEDGED` ersetzt und kann in kommenden Releases ggf. veraltet sein.

- `MAJORITY`: Die Operation wartet auf die Mehrheit der Server, bevor sie ausgeführt wird. Exceptions werden sowohl bei Netzwerkproblemen als auch bei Serverfehlern ausgeworfen.
- `FSYNC_SAFE`: Die Operation wartet, bis der Server sichergestellt hat, dass die Daten auf der Festplatte angekommen sind. Exceptions werden sowohl bei Netzwerkpoblemen als auch bei Serverfehlern ausgeworfen.

  > Dieser Write Concern wurde durch den Write Concern `FSYNCED` ersetzt und kann in kommenden Releases ggf. veraltet sein.

- `JOURNAL_SAFE`: Die Operation wartet, bis der Server die Daten zu den Journal-Files committet. Exceptions werden sowohl bei Netzwerkproblemen als auch bei Serverfehlern ausgeworfen.

  > Dieser Write Concern wurde durch den Write Concern `JOURNALED` ersetzt und kann in kommenden Releases ggf. veraltet sein.

- `REPLICAS_SAFE`: Die Operation wartet auf mindestens zwei Server, bevor sie ausgeführt wird. Exceptions werden sowohl bei Netzwerkproblemen als auch bei Serverfehlern ausgeworfen.

> Dieser Write Concern wurde durch den Write Concern `REPLICA_ACKNOWLEDGED` ersetzt und kann in kommenden Releases ggf. veraltet sein.

> Zusätzliche Informationen zu den Write Concerns finden Sie unter:
> - http://docs.mongodb.org/manual/core/write-concern/#write-operations-write-concern
> - http://docs.mongodb.org/ecosystem/drivers/java-replica-set-semantics/
> - http://api.mongodb.org/java/current/com/mongodb/WriteConcern.html#field_summary

Statuswerte

`1`	Die Operation wurde erfolgreich ausgeführt.
`0`	Die Operation wurde erfolgreich ausgeführt, jedoch ohne Ergebnis.
`-1`	Die Operation ist aufgrund eines technischen Fehlers fehlgeschlagen.

10.24.4.2 Verbindung konfigurieren

Die Verbindung zur *MongoDB* wird in der `mongodb-connections.xml` innerhalb des X4-Installationsordners konfiguriert, z. B. `X4\Server\X4DB\0\mongodb-connections.xml`.

```
<MongoDB>
    <!-- connects to localhost on the default port without authentication -->
    <Connection name="#default" db="test"/>

    <Connection name="localhost" db="test">
        <Server host="localhost" port="27017"/>
    </Connection>
</MongoDB>
```

Erläuterung:

- `/MongoDB/Connection`: Stellt eine benannte Verbindung dar. Kann mehrfach existieren.
- `/MongoDB/Connection/@name`: Alias/Referenzname für die Verbindung
- `/MongoDB/Connection/@db`: Name der Datenbank zu der die Verbindung hergestellt werden soll, d.h., Datenbanken müssen immer über externe Werkzeuge (z .B. *MongoShell*) angelegt werden.

- `/MongoDB/Connection/Server`: Stellt einen Servereintrag für eine Verbindung dar. Kann mehrfach pro Connection existieren, um einen Cluster zu adressieren.
- `/MongoDB/Connection/Server/@host`: IP/DNS-Name des Servers, zu dem eine Verbindung hergestellt werden soll (`default: 127.0.0.1`).
- `/MongoDB/Connection/Server/@port`: Port des Servers, zu dem eine Verbindung hergestellt werden soll (`default: 27017`).
- `/MongoDB/Connection/Credential`: Anmeldedaten für eine Verbindung. Kann mehrfach pro Verbindung existieren.
- `/MongoDB/Connection/Credential/@user`: Benutzername der Anmeldung
- `/MongoDB/Connection/Credential/@password`: Passwort der Anmeldung
- `/MongoDB/Connection/Credential/@source`: Ursprung der Anmeldedaten (Nutzung zZt. unklar)

> **Beachten Sie:**
> Wenn eine Verbindung keine Serverelemente besitzt, wird unabhängig davon, ob Credential-Elemente existieren, ohne Authentifizierungsdaten eine Verbindung mit `localhost:27017` hergestellt.

10.24.4.3 Kodierung einer Dokumentenstruktur

Die *JSON/Mongo*-Dokumente werden nach einfachen Regeln in ihr XML-Äquivalent übersetzt:

- Dokumente und Objekte/Substrukturen werden durch `<Object>` dargestellt. Die Schlüssel-Wert-Einträge eines Objekts werden als Kindelemente des `Object`-Elements abgebildet.
- Arrays werden durch `<Array>` dargestellt. Die Array-Einträge werden als Kindelemente des `Array`-Elements abgebildet.
- Wenn es kein Array oder Object ist, wird es durch `<Value>` dargestellt.
- Der Datentyp eines Values wird in dem Attribut `type` auf dem `Value`-Element festgehalten (siehe Unterstützte Datentypen). Wenn kein Datentyp angegeben ist, wird `String` angenommen.
- Die Schlüssel-Wert-Einträge eines Objektes oder eine Substruktur bekommen den Schlüssel als `name`-Attribute auf das Element des Wert-Elements gesetzt.

Invarianten:

- Für jede `{` im JSON wird ein `Object`-Element im XML angelegt. (Ausnahme: Die Dokumentenwurzel wird in Kommandos durch andere Elemente abgebildet.)
- Für jede `[` im JSON wird ein `Array`-Element im XML angelegt.
- Für jeden `:` im JSON (der nicht Bestandteil eines String-Literals ist) wird ein `name`-Attribut angelegt, das den Wert vor dem `:` als Inhalt enthält.
- Alle Kindelemente eines `Object`-Elements haben ein `name`-Attribut.
- Kindelemente eines `Array`-Elements haben kein `name`-Attribut.
- `Value`-Elemente haben kein Kindelement.

- `Text`-Knoten existieren nur als Kinder von `Value`-Elementen (Ausnahme: formatierender Whitespace).
- `type`-Attribute existieren nur auf `Value`-Elementen (aber nicht jedes `Value`-Element muss ein `type`-Attribute haben).

Beispiel

```
<Object>
    <Value name="some" type="String">thing</Value>
</Object>
```

Die Dokumenten-Wurzel kann auch durch andere Elemente ersetzt werden, um die Erzeugung eines `Object`-Elements zu vermeiden. In diesem Fall wäre dort immer ein `Object`-Element notwendig.

10.24.4.4 Unterstützte Datentypen

Der Adapter unterstützt folgende Datentypen:

- `Null`
- `ObjectId`
- `Boolean`
- `Long`
- `Int`
- `Double`
- `String`
- `DateTime`
- `Integer`: String wird als `BigInteger` gelesen und anschließend als `Long` gespeichert.
- `Decimal`: String wird als `BigDecimal` gelesen und anschließend als `Double` gespeichert.
- `Numeric`: Wenn der String einen `.` enthält, wird er als `Decimal` gelesen, sonst als `Integer`.
- `RegEx`: regulärer Ausdruck gemäß http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html

Dieses `Value`-Element kennt zudem das Attribut `flags`, welches einen String aus folgenden Zeichen ist und die entsprechende Option auf dem Muster setzt:

- `c: CANON_EQ`
- `i: CASE_INSENSITIVE`
- `m: MULTILINE`
- `s: DOTALL`
- `u: UNICODE_CASE`
- `L: LITERAL`
- `N: UNIX_LINES`
- `U: UNICODE_CHARACTER_CLASS`
- `Z: COMMENTS`

10.24.4.5 Input

Der *MongoDB Adapter* erwartet spezifische Input-XML-Strukturen. Diese Input-XML-Strukturen enthalten eine Wurzel mit mehreren Kommandos darunter. Ebenso wird auch das `resulttag`-Attribut unterstützt.

Input-Struktur

```
<!-- Ein Root-Element mit beliebigem Namen. -->
<root>
    <insert collection="products" resulttag="insertOutput">
        <Value name="item" type="String">card</Value>
        <Value name="qty" type="Integer">15</Value>
    </insert>
    <find collection="products" resulttag="findOutput">
        <query>
            <Value name="item" type="String">card</Value>
        </query>
    </find>
    <!-- Hier können weitere Kommandos angegeben werden. -->
</root>
```

Dies ist ein vollständiges und ausführbares Beispiel. Alle nachfolgenden Beispiele sind lediglich Ausschnitte, die zur Ausführung in eine vollständige Input-Struktur mit einem umschließenden Root-Element eingefügt werden müssen.

Unterstützte Kommandos

- `copy` (kein MongoDB-Kommando)
- `find`
- `insert`
- `remove`
- `save`

- update

10.24.4.5.1 Kommando copy

Mit copy werden alle Kindelemente des copy-Kommandos in die Ausgabe kopiert.

Input

Das copy-Kommando hat folgende Syntax:

```
<copy resulttag="output">
    <Child1/>
    <Child2>
        <Grandchild1/>
        <Grandchild2/>
    </Child2>
</copy>
```

Output

Dieses Kommando gibt folgende XML-Struktur aus:

```
<output>
    <Child1/>
    <Child2>
        <Grandchild1/>
        <Grandchild2/>
    </Child2>
</output>
```

10.24.4.5.2 Kommando insert

Mit insert wird ein Dokument in die Collection eingefügt. Das insert-Element ist dabei gleichzeitig die Dokumentenwurzel.

- Bitte beachten Sie, dass Batchinsert mit insert nicht unterstützt wird.
- Weitere Informationen zu diesem Kommando finden Sie unter http://docs.mongodb.org/manual/reference/method/db.collection.insert/#db.collection.insert.

Input

Das insert-Kommando hat folgende Syntax:

```
<insert collection="foo">
    <Value name="bar" type="String">baz</Value>
</insert>
```

Output

Dieses Kommando gibt ein `WriteResult`-Objekt aus, das den Status der Operation enthält.

10.24.4.5.3 Kommando `save`

Mit `save` wird gemäß mitgegebenen Daten ein neues Dokument angelegt oder ein bestehendes Dokument aktualisiert. Das `save`-Element ist dabei gleichzeitig die Dokumentenwurzel.

> Weitere Informationen zu diesem Kommando finden Sie unter http://docs.mongodb.org/manual/reference/method/db.collection.save/#db.collection.save.

Input

Das `save`-Kommando hat folgende Syntax:

```
<save collection="foo">
    <Value name="bar" type="String">baz</Value>
</save>
```

Output

Dieses Kommando gibt ein `WriteResult`-Objekt aus, das den Status der Operation enthält.

10.24.4.5.4 Kommando `update`

Mit `update` wird ein Dokument in einer Collection modifiziert. Je nach `update`-Parameter kann die Methode spezielle Felder eines bereits bestehenden Dokuments oder ein komplettes Dokument modifizieren.

> Weitere Informationen zu diesem Kommando finden Sie unter http://docs.mongodb.org/manual/reference/method/db.collection.update/#db.collection.update.

Input

Das `update`-Kommando hat folgende Syntax:

```
<update collection="foo">
    <query/>
    <update upsert="false" multi="false"/>
</update>
```

Erläuterung:

- Das `query`-Element enthält den Abfrage/Selektionsteil des `update`-Elements und ist auch Dokumentenwurzel für die Abfrage.

- Das `update`-Element enthält den Veränderungsteil des Updates und ist auch Dokumentenwurzel für die Veränderung.
- Die Optionen `upsert` und `multi` werden auf dem inneren `update`-Element angegeben.

Output

Dieses Kommando gibt ein `WriteResult`-Objekt aus, das den Status der Operation enthält.

10.24.4.5.5 Kommando `remove`

Mit `remove` wird ein Dokument aus einer Collection entfernt.

> Weitere Informationen zu diesem Kommando finden Sie unter http://docs.mongodb.org/manual/reference/method/db.collection.remove/#db.collection.remove.

Input

Das `remove`-Kommando hat folgende Syntax:

```
<remove collection="foo">
    <query/>
</remove>
```

Erläuterung:

- Das `query`-Element enthält den Abfrage/Selektionsteil des `remove`-Elements und ist auch Dokumentenwurzel für die Abfrage.

> Beachten Sie, dass die Option `justOne` durch den Java-Treiber nicht unterstützt wird.

Output

Dieses Kommando gibt ein `WriteResult`-Objekt aus, das den Status der Operation enthält.

10.24.4.5.6 Kommando `find`

Mit

`find` wird ein Dokument aus einer Collection selektiert.

> Weitere Informationen zu diesem Kommando finden Sie unter http://docs.mongodb.org/manual/reference/method/db.collection.find/#db.collection.find.

Input

Das `find`-Kommando hat folgende Syntax:

```
<find collection="foo">
<--! Optional elements -->
  <query/>
  <projection/>
    <sort/>
    <hint/>
    <min/>
    <max/>
    <special/>
    <cursor skip="0" limit="0" batchSize="0"/>
</find>
```

Erläuterung:

- Das `query`-Element enthält den Abfrage/Selektionsteil des `find`-Elements und ist auch Dokumentenwurzel für die Abfrage.
- Das `projection`-Element enthält die Feldauswahl des `find`-Elements und ist auch Dokumentenwurzel für die Auswahl.

Die weiteren optionalen Elemente steuern das Verhalten des Ergebnis-Cursors:

- `sort` (Dokumentenwurzel), siehe auch http://docs.mongodb.org/manual/reference/method/cursor.sort/#cursor.sort
- `hint` (Dokumentenwurzel), siehe auch http://docs.mongodb.org/manual/reference/method/cursor.hint/#cursor.hint
- `min` (Dokumentenwurzel), siehe auch http://docs.mongodb.org/manual/reference/method/cursor.min/#cursor.min
- `max` (Dokumentenwurzel), siehe auch http://docs.mongodb.org/manual/reference/method/cursor.max/#cursor.max
- `special` (quasi-Dokumentenwurzel), siehe auch http://api.mongodb.org/java/current/com/mongodb/DBCursor.html#addSpecial(java.lang.String,%20java.lang.Object)

  > ⓘ Für jeden Kindknoten in der Wurzel des `special`-Dokuments wird die `addSpecial`-Methode aufgerufen.

- `cursor` (no body) ist lediglich der Träger für folgende drei Attribute:
  - `skip`, siehe auch http://docs.mongodb.org/manual/reference/method/cursor.skip/#cursor-skip
  - `limit`, siehe auch http://docs.mongodb.org/manual/reference/method/cursor.limit/#cursor.limit
  - `batchSize`, siehe auch http://docs.mongodb.org/manual/reference/method/cursor.batchSize/#cursor.batchSize

Output
Dieses Kommando gibt einen Cursor aus, der auf die Dokumente zeigt, welche die `query`-Kriterien erfüllen.

10.24.4.6 Beispiele

Folgende Beispiele zeigen Mongo-Ausdrücke und ihre jeweiligen Entsprechungen in XML.

Beispiele für die Dokumentenstruktur:

JSON/Mongo-Struktur	Entsprechende XML-Struktur
`{}`	`<Object/>`

`{ type: "snacks" }`	`<Object>`
	`  <Value name="type" type="String">snacks</Value>`
	`</Object>`

`{ type: { $in: [ 'food', 'snacks' ] } }`	`<Object>`
	`  <Object name="type">`
	`  <Array name="type">`
	`  <Value type="String">food</Value>`
	`  <Value type="String">snacks</Value>`
	`  </Array>`
	`  </Object>`
	`</Object>`

`{ type: 'food', price: { $lt: 9.95 } }`	`<Object>`
	`  <Value name="type" type="String">food</Value>`
	`  <Object name="price">`
	`  <Value name="$lt" type="Double">9.95</Value>`
	`  </Object>`
	`</Object>`

JSON/Mongo-Struktur	Entsprechende XML-Struktur
`{ $or: [ { qty: { $gt: 100 } }, { price: { $lt: 9.95 } } ] }`	`<Object> <Array name="$or"> <Object> <Object name="qty"> <Value name="$gt" type="Integer">100</Value> </Object> </Object> <Object> <Object name="price"> <Value name="$lt" type="Double">9.95</Value> </Object> </Object> </Array> </Object>`

Beispiele für Aufrufe:

Mongo-Aufruf	Entsprechende XML-Struktur
`db.products.insert( { item: "card", qty: 15 } )`	`<insert collection="products"> <Value name="item" type="String">card</Value> <Value name="qty" type="Integer">15</Value> </insert>`
`db.inventory.save( { type: "book", item: "notebook", qty: 40 } )`	`<save collection="inventory"> <Value name="type" type="String">book</Value> <Value name="item" type="String">notebook</Value> <Value name="qty" type="Integer">40</Value> </save>`

Mongo-Aufruf	Entsprechende XML-Struktur
`db.books.update( { item: "The Banquet" }, { item: "The Banquet", price: 19 , stock: 3 } )`	`<update collection="books"> <query> <Value name="item" type="String">The Banquet</Value> </query> <update upsert="true"> <Value name="item" type="String">The Banquet</Value> <Value name="price" type="Double">19< /Value> <Value name="stock" type="Integer">3< /Value> </update> </update>`
`db.books.update( { item: "Divine Comedy" }, { $set: { price: 18 }, $inc: { stock: 5 } } )`	`<update collection="books"> <query> <Value name="item" type="String">Div ine Comedy</Value> </query> <update upsert="true"> <Object name="$set"> <Value name="price" type="Double">19</Value> </Object> <Object name="$inc"> <Value name="stock" type="Integer">5</Value> </Object> </update> </update>`
`db.inventory.update( { type: "book", item : "journal" }, { $set : { qty: 10 } }, { upsert : true } )`	`<update collection="inventory"> <query> <Value name="type" type="String">boo k</Value> <Value name="item" type="String">jou rnal</Value> </query> <update upsert="true"> <Object name="$set"> <Value name="qty" type="Integer">1 0</Value> </Object> </update> </update>`

Mongo-Aufruf	Entsprechende XML-Struktur
`db.inventory.update( { type : "book" }, { $inc : { qty : -1 } }, { multi: true } )`	`<update collection="inventory"> <query> <Value name="type" type="String">book</Value> </query> <update multi="true"> <Object name="$inc"> <Value name="qty" type="Integer">-1</Value> </Object> </update> </update>`
`db.inventory.find()`	`<find collection="inventory" />`
`db.inventory.find({})`	`<find collection="inventory"> <query/> </find>`
`db.inventory.find( { type: "snacks" } )`	`<find collection="inventory"> <query> <Value name="type" type="String">snacks</Value> </query> </find>`

Mongo-Aufruf	Entsprechende XML-Struktur

`db.inventory.find( { type: 'food' }, { _id: 0, item: 1, qty: 1 } )`	`<find collection="inventory">`
	`  <query>`
	`  <Value name="type" type="String">food</Value>`
	`  </query>`
	`  <projection>`
	`  <Value name="_id" type="Integer">0</Value>`
	`  <Value name="item" type="Integer">1</Value>`
	`  <Value name="qty" type="Integer">1</Value>`
	`  </projection>`
	`</find>`
	oder
	`<find collection="inventory">`
	`  <query>`
	`  <Value name="type" type="String">food</Value>`
	`  </query>`
	`  <projection>`
	`  <exclude name="_id" />`
	`  <include name="item" />`
	`  <include name="qty" />`
	`  <!-- excludes und includes können mit Object, Array und Value im projection-Element gemischt werden -->`
	`  </projection>`
	`</find>`

Mongo-Aufruf	Entsprechende XML-Struktur

`db.stocks.find().sort( { ticker: 1, date: -1 } )`	`<find collection="stocks"><sort><Value name="ticker" type="Integer">1</Value><Value name="date" type="Integer">-1</Value></sort></find>`
	oder
	`<find collection="stocks"><sort><ascending name="ticker" /><descending name="date" /><!-- ascending und descending können mit Object, Array und Value im sort-Element gemischt werden --></sort></find>`

`db.stocks.find().sort( { ticker: 1 } ).limit(100)`	`<find collection="stocks"><sort><ascending name="ticker" /></sort><cursor limit="100"/></find>`
`db.stocks.find().sort( { ticker: 1 } ).skip(200).limit(100)`	`<find collection="stocks"><sort><ascending name="ticker" /></sort><cursor skip="200" limit="100"/></find>`

10.24.5 Repository Explorer (veraltet)

`Repository Explorer`: Listet Dateien und Ordner eines bestimmten Ordnerpfads im X4 Repository (z.B. im Ordner `C:\X4\X4DB`) oder des gesamten Repositorys.
Der Adapter gibt ein XML-Dokument aus, das jeweils die Datei- und Ordnernamen auflistet – je nach gewählter Operation lassen sich die Ordnerinhalte auch rekursiv (einschließlich Unterordnern)

auflisten. Optional kann zudem der Inhaltstyp ermittelt werden, wobei bei XML-Dateien der jeweilige Wurzel-Elementname mit ausgegeben wird.

⚠ Der *Repository Explorer* wird mit Version 6.2.0 der X4 BPMS abgekündigt und in zukünftigen Versionen der X4 BPMS nicht mehr unterstützt. Bitte verwenden Sie stattdessen den Project Access oder Projects Adapter.

ⓘ Wenn über Parameter `DiscoverContent` der Inhaltstyp für die Repository-Dateien ermittelt wird, kann dies bei umfangreicheren Ordnerstrukturen mit zahlreichen Dateien längere Zeit in Anspruch nehmen!

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`List`*: Datei- und Ordnernamen auflisten, die sich innerhalb von StartPath befinden • *`DeepListFlat`*: Datei- und Ordnernamen einschließlich aller Unterordner auflisten, die sich innerhalb von `StartPath` befinden • *`DeepListNested`*: wie *`DeepListFlat`*; das erzeugte XML-Dokument bildet jedoch die Ordnerstruktur hierarchisch ab • *`DeepListFlatFromRoot`*: Datei- und Ordnernamen des gesamten X4 Repositorys (X4DB) einschließlich aller Unterordner auflisten • *`DeepListNestedFromRoot`*: wie *`DeepListFlatFromRoot`*; das erzeugte XML-Dokument bildet jedoch die Ordnerstruktur hierarchisch ab

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.repository.RepositoryAdapter`*: Hauptklasse (Standard)
StartPath	Pfad im Repository **Mögliche Werte:** Beliebige Pfadangabe zum Repository-Ordner einschließlich des Benutzerordners (z.B. *`1/Project/Folder`*)

DiscoverContent	Inhaltstyp der Repository-Dateien prüfen (XML oder nicht-XML) **Mögliche Werte:** • *Ja*: Inhaltstyp von Repository-Dateien prüfen und im Ergebnisdokument jedem Element `File` ein Kindelement mit dem ermittelten Inhaltstyp hinzugefügt (kostet Performance!) • *Nein*: Keine Inhaltsprüfung (Standard)

Input

Dieser Adapter benötigt keinen Input.

Ausgabe

Der Adapter gibt je nach Operation ein XML-Dokument nach folgendem Muster aus, das Ordner in jeweils einem Element `<Directory>` und Dateien jeweils in einem Element `<File>` auflistet:

```
<Repository repositoryPath="Absoluter Repository-Pfad im Dateisystem">

  <!-- Bei Ordnern: -->
  <Directory name="Ordnername"
   path="Repository-Pfad des Ordners inkl. User Collection"
   fullpath="Absoluter Repository-Pfad des Ordners im Dateisystem" />

  <!-- Bei Dateien: -->
  <File name="Dateiname"
   path="Repository-Pfad der Datei inkl. User Collection"
   fullpath="Absoluter Repository-Pfad der Datei im Dateisystem">

   <!-- falls 'DiscoverContent' aktiviert ist: -->
   <Inhaltstyp: XML oder Non-XML>
     <!-- falls Inhaltstyp XML ist: -->
     <Wurzel-Elementname />
   </Inhaltstyp: XML oder Non-XML>
  </File>
</Repository>
```

Beispiel

Im folgenden Beispiel wird von einem Ordner `Folder` mit Operation `DeepListNested` die Ordnerstruktur hierarchisch ausgegeben und über Parameter `DiscoverContent` der Inhaltstyp von Dateien ermittelt. Für die XML-Datei `File.xml` wird hierbei dessen Wurzel-Element ausgegeben:

```
<Repository repositoryPath="c:\X4\X4DB">
  <Directory name="Folder" path="/1/Project/Data"
  fullpath="c:/X4/X4DB/1/Project/Data">
   <Directory name="EmptySubfolder" path="/1/Project/Data/Folder"
    fullpath="c:/X4/X4DB/1/Project/Data/Folder"/>
   <File name="File.txt" path="/1/Project/Data/Folder"
    fullpath="c:/X4/X4DB/1/Project/Data/Folder">
     <Non-XML/>
   </File>
   <File name="File.xml" path="/1/Project/Data/Folder"
    fullpath="c:/X4/X4DB/1/Project/Data/Folder">
     <XML><RootElement/></XML>
   </File>
  </Directory>
</Repository>
```

10.24.6 JSON Converter (veraltet)

`JSON Converter` : Konvertiert ein Textdokument, das strukturierte Daten in der *JavaScript Object Notation (JSON)* enthält, in ein XML-Dokument mit dem Wurzel-Element `<JSON>`, und konvertiert umgekehrt ein XML-Dokument in ein UTF-8-kodiertes JSON-Dokument.

- Veraltet: Bitte den JSON Converter 2.0 verwenden!
- Ehemalige Bezeichnung bis Version 4.5.1: *JSON Converter Adapter*

Bitte beachten Sie bei der Konvertierung von XML ins JSON-Format:

- Processing Instructions, Kommentare, Entities und CDATA-Abschnitte im Input-XML-Dokument können nicht konvertiert werden.
- Wenn Sie aus XML-Elementstrukturen explizit Arrays in JSON erzeugen möchten, fügen Sie dem entsprechenden Element das Attribut `class="array"` hinzu.
- Um leere Arrays in JSON zu unterdrücken, fügen Sie dem entsprechenden XML-Element ein Attribut `class="object"` hinzu, so wird beispielsweise `<element class="object"/>` in JSON `element : null`.
- Mehr Informationen zur verwendeten Bibliothek finden Sie unter http://json-lib.sourceforge.net/usage.html.

Eigenschaften

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** • *`JSON2XML`*: JSON-Dokument in XML konvertieren • *`XML2JSON`*: XML-Dokument in JSON konvertieren

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.json.JSONConverter`*: Hauptklasse (Standard)
classicMode	Legt fest, dass die gleichen XML- und JSON-Strukturen wie in älteren X4 Versionen erstellt werden. **Mögliche Werte:** `true`: Der Klassische Modus wird verwendet `false`: Der Klassische Modus wird nicht verwendet (Standard)

10.24.7 ZIP Compression (veraltet)

`ZIP Compression`: Erzeugt ein ZIP-Archiv mit einer oder mehreren Dateien aus Daten, die in einer adapterspezifischen XML-Struktur abgelegt sind, und speichert das ZIP-Archiv in einem lokalen Zielordner.

> ⚠ Der *ZIP Compression* Adapter wird mit Version 6.2.0 der X4 BPMS abgekündigt und in zukünftigen Versionen der X4 BPMS nicht mehr unterstützt. Bitte verwenden Sie stattdessen den Zip Converter.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *`Zip`*: ZIP-Archiv aus den Daten im Input-XML-Dokument erzeugen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.zipadapter.ZipAdapter*: Hauptklasse (Standard)
directory	Zielordner, in den das ZIP-Archiv gespeichert werden sollen ⓘ Ab Version 5.0 der X4 BPMS ist ein abschließender Schrägstrich nicht mehr erforderlich, um das Verzeichnis vom ZIP-Namen zu trennen. **Mögliche Werte:** Beliebige gültige Zeichenkette (z. B. *%USER_COLLECTION%\Project\Folder*)
zipName	Dateiname des erzeugten ZIP-Archivs **Mögliche Werte:** Beliebiger gültiger Dateiname (z. B. *`Output.zip`*)

Input

Dieser Funktions-Adapter erwartet eine adapterspezifische Input-XML-Struktur. Diese enthält für jede Datei, die im ZIP-Archiv enthalten sein soll, den Dateinamen und den Base64-kodierten Dateiinhalt. Jedes Element `<document/>` steht hierbei für eine Datei im ZIP-Archiv.

```
<data>
    <documents>
        <document>
            <filename><!--Dateiname--></filename>
            <content><!--Base64-kodierte Daten--></content>
        </document>
    </documents>
</data>
```

10.24.8 ZIP Folder Compression (veraltet)

`ZIP Folder Compression`: Erzeugt ein ZIP-Archiv von einem Ordner und gibt das ZIP-Archiv als Binärdokument mit MIME-Typ *`Application/ZIP`* aus.

⚠ Der *ZIP Folder Compression* Adapter wird mit Version 6.2.0 der X4 BPMS abgekündigt und in zukünftigen Versionen der X4 BPMS nicht mehr unterstützt. Bitte verwenden Sie stattdessen den Zip Converter.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Funktions-Adapter durchführt **Mögliche Werte:** *Execute*: ZIP-Archiv vom gewählten Ordner erzeugen

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *de.softproject.integration.adapter.zipfolderadapter.ZipFolderAdapter*: Hauptklasse (Standard)
zipFolderPath	Pfad zum Ordner, von dem ein ZIP-Archiv erstellt werden soll **Mögliche Werte:** • Beliebige gültige Zeichenkette (z.B. *%USER_COLLECTION%\Project\Folder*) • Pfad ohne */* bzw. ** am Ende: Das erzeugte ZIP-Archiv enthält als einzigen Unterordner den gewählten Ordner, und darin dessen Ordnerinhalte • Pfad mit */* bzw. ** am Ende: Das erzeugte ZIP-Archiv enhält nur die Inhalte des gewählten Ordners, nicht jedoch den Ordner selbst • *(nicht gesetzt)*: Es wird versucht, aus einem Input-XML-Dokument eine Zeichenkette zu ermitteln, die über den XPath-Ausdruck `/X4/ZipFolderAdapter/zipfolderpath` adressiert wurde

10.24.9 ZIP in Memory Compression (veraltet)

`ZIP in Memory Compression`: Erzeugt ein ZIP-Archiv mit einer oder mehreren Dateien aus Daten, die in einer adapterspezifischen XML-Struktur abgelegt sind, und hält die Daten komplett im Arbeitsspeicher – benötigt also keinen Zugriff auf das Dateisystem.

> ⚠ Der *ZIP in Memory Compression* Adapter wird mit Version 6.2.0 der X4 BPMS abgekündigt und in zukünftigen Versionen der X4 BPMS nicht mehr unterstützt. Bitte verwenden Sie stattdessen den Zip Converter.

Eigenschaften (Properties)

Operation	Bestimmt, welche Operation der Adapter durchführt **Mögliche Werte:** • *`Zip`*: ZIP-Archiv aus den Daten im Input-XML-Dokument im Arbeitspeicher erzeugen • *`Unzip`*: ZIP-Archiv in den Arbeitsspeicher entpacken

Parameter

Adapter	Hauptklasse des Adapters (nicht verändern!) **Mögliche Werte:** *`de.softproject.integration.adapter.zipinmemory.ZipInMemory`*: Hauptklasse (Standard)
encoding	Zeichenkodierung für Datei-/Pfadnamen **Mögliche Werte:** • Beliebige gültige Zeichenkodierung (z.B. *`UTF-8`*) • *`CP850`*: MS-DOS Deutschland, wird von *MS Windows* so erzeugt (Standard) • *`(leer)`*: UTF-8 (aus Abwärtskompatibilitäts-Gründen mit älteren *X4 Server*-Versionen

Input

Wenn die Eigenschaft `Operation` auf *`Zip`* gesetzt ist, erwartet dieser Adapter eine adapterspezifische Input-XML-Struktur. Diese enthält für jede Datei, die im ZIP-Archiv enthalten sein soll, den Dateinamen und den Base64-kodierten Dateiinhalt. Jedes Element `<File>` steht hierbei für eine Datei, jedes Element `<Directory>` für einen leeren Ordner im ZIP-Archiv.

```
<Archiv>
    <File name="Dateiname" comment="Optionaler Kommentar">
        <!--Base64-kodierte Daten-->
    </File>
    <Directory name="Ordnername/" />
</Archiv>
```

Ausgabe

Wenn die Eigenschaft `Operation` auf *`Unzip`* gesetzt ist, erzeugt dieser Funktions-Adapter aus einem ZIP-Archiv eine adapterspezifische XML-Struktur, beispielsweise:

```
<Archiv>
    <File name="Data.txt">Q29udGVudCBvZiB0aGlzIHRleHQgZmlsZQ==</File>
    <Directory name="Folder/" />
</Archiv>
```

Erläuterung:

- Jedes Element `<Directory>` steht für einen (leeren) Ordner im ZIP-Archiv.
- Jedes Element `<File>` steht für eine Datei und speichert den Base64-kodierten Dateiinhalt als Wert.
- Attribut `name` gibt den Datei- oder Ordnernamen an.
- Attribut `size` gibt die Dateigröße an, sofern diese im ZIP-Archiv hinterlegt ist.