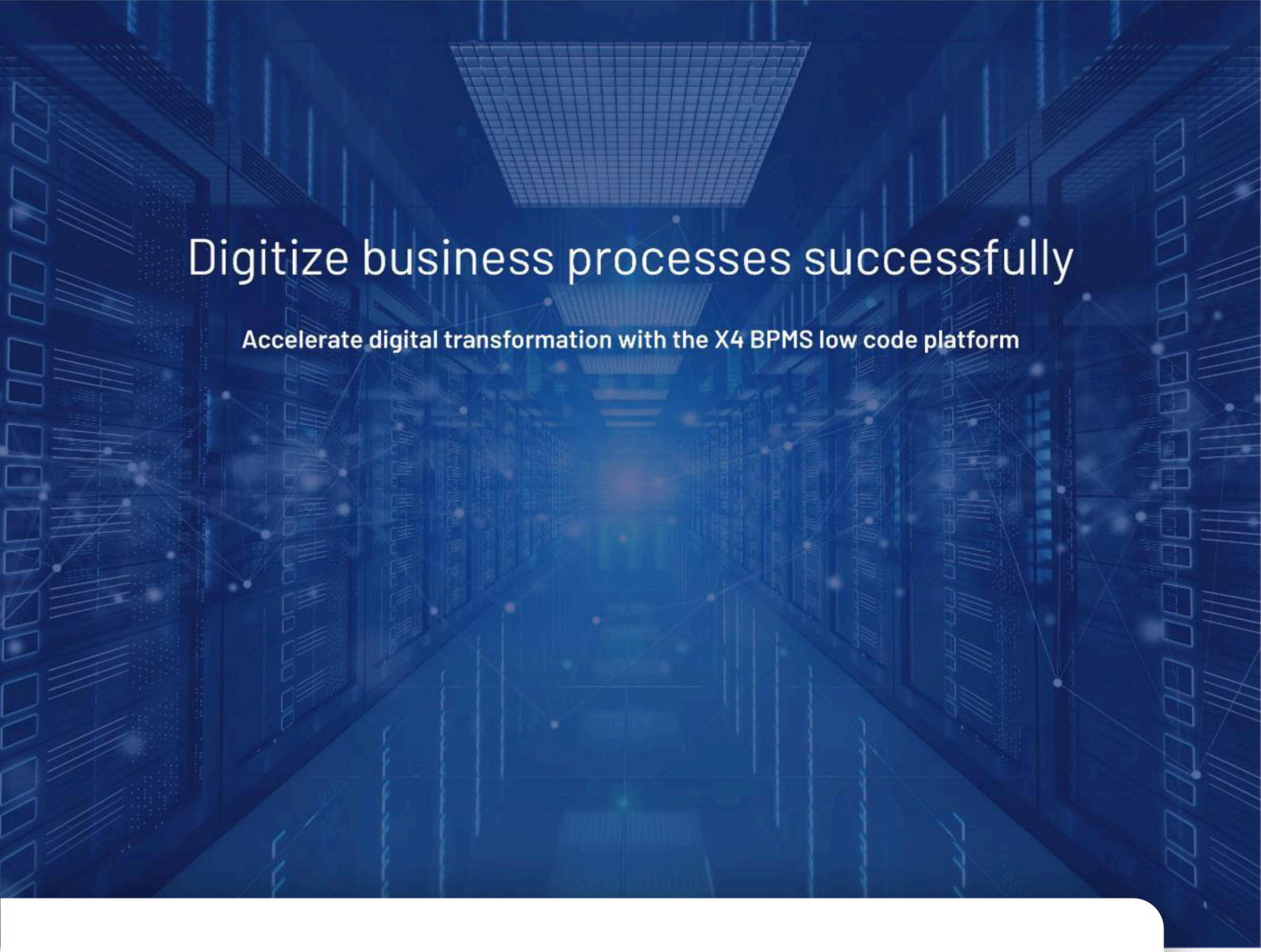


X4 ESB

SoftProject

The information in this document is subject to change without notice. SoftProject GmbH assumes no responsibility for any errors that may appear in this document.

This document may not be copied, photocopied, reproduced, translated or converted to any electronic or machine-readable form in whole or in part without prior written approval of SoftProject GmbH.

Mentioned products are trademarks or registered trademarks of their respective owners.

Contact

SoftProject GmbH

Am Erlengraben 3

D-76275 Ettlingen – Germany

Website: www.softproject.de

Sales

Phone: +49 7243 56175-0

vertrieb@softproject.de

SoftProject Support

Phone: +49 7243 56175-333

support@softproject.de

© SoftProject GmbH. All rights reserved.

Table of Contents

1 Introduction 23
1.1 What is an ESB Project? 23
1.2 How can I use ESB Projects? 23
2 Interface 24
2.1 Toolbar 24
2.2 Context Menu 26
2.3 Perspectives 29
2.3.1 Switching perspectives 29
2.3.2 Reseting perspectives 29
2.4 Views 29
2.4.1 Open views 35
2.5 Editors 35
2.5.1 BPMN Editor 36
2.5.2 Database Editor 36
2.5.2.1 Statement Editor 37
2.5.2.2 Mapping Editor 39
2.5.3 JSON Editor 40
2.5.4 Mapping Editor 41
2.5.4.1 Shortcuts in the Mapping Editor 44
2.5.5 Process Editor 46
2.5.5.1 Process Quality Index 46
2.5.5.2 Shortcuts in the Process Editor 47
2.5.6 XML Editor 49
3 Process Components and Their Use 51
3.1 Connecting Process Components 52
3.2 Control Components 53
3.2.1 Stop 53
3.2.2 Fork and Join 54
3.2.3 Merge 56

3.2.4 Condition 57
3.2.4.1 Creating and Using Conditions 58
3.2.4.2 Condition Editor 66
3.2.5 Connection 70
3.2.6 Pause 71
3.2.6.1 Dynamically Parameterizing the Pause Component 72
3.2.7 Counter 73
3.2.8 Transaction 74
3.2.9 Variable 75
3.2.9.1 Using Variables 76
3.2.9.2 Context Variables 77
3.3 Module Components 78
3.3.1 Adapter 78
3.3.2 XML 81
3.3.2.1 XML Editor 82
3.3.3 JSON Document 83
3.3.3.1 JSON Editor 85
3.3.4 Document 86
3.3.4.1 ASCII Descriptor 87
3.3.5 Subprocess 98
3.3.5.1 Using private subprocesses 99
3.3.6 Mapping 100
3.3.6.1 Creating and Using an XSL Mapping 101
3.3.6.2 Mapping Editor 105
3.3.7 PDF Creator 108
3.4 Drawing Elements 110
3.4.1 Connector 110
3.4.2 Rectangle and Ellipse 112
3.4.3 Textbox 113
3.4.4 Polygon 114
3.4.5 Polyline and Line 114

3.4.6 Image ... 115
4 Working with ESB Projects ... 116
4.1 Templates ... 116
4.2 The Structure of ESB Projects ... 116
4.3 Creating ESB Projects ... 116
4.4 Managing Repository Projects ... 117
4.4.1 Import repository elements ... 117
4.4.1.1 Import via drag and drop ... 118
4.4.1.2 Using the Import Wizard for files and projects ... 118
4.4.2 Export repository elements ... 118
4.4.2.1 Using the Export Wizard for files and projects ... 119
4.4.3 Renaming elements and updating references automatically ... 119
4.4.4 Moving elements and updating references automatically ... 120
4.4.5 Repairing Repository References ... 121
4.4.6 Deleting Unused Resources ... 122
4.4.7 Searching Repository Elements ... 123
4.4.8 Migrating Outdated Projects ... 124
5 Creating and Modelling Processes ... 126
5.1 Creating Processes ... 126
5.1.1 Using Private Files Below Processes ... 127
5.2 Properties of a Technical Process (.wrf) ... 128
5.3 Aligning Process Components ... 130
5.4 Validating Processes ... 131
5.4.1 Checking Processes ... 131
5.4.2 Default Validation Rules ... 132
5.4.3 Custom Validation Rules ... 132
5.5 Parameterizing Process Components ... 134
5.5.1 Parameterization Options ... 134
5.5.2 Use Cases for Parameters ... 138
5.5.3 Setting Dynamic Parameters ... 139
5.5.3.1 Setting Parameters Dynamically via XSL Mappings ... 140

5.5.3.2 Parameterizing Dynamically without Data Transformation 140
5.5.3.3 Dynamic Parameters with DynamicParameter 141
5.5.4 Using Component Parameters 142
5.5.4.1 Defining Component Parameters 143
5.5.5 Setting custom states with ConditionState 144
5.5.6 Using Parameters in XSL Mappings 145
5.6 Creating and Using Parameters 147
5.6.1 System Parameters 147
5.6.1.1 Overview of System Parameters 148
5.6.1.2 Using System Parameters 152
5.6.2 Global Parameters and Groups 153
5.6.2.1 Persisting Global Parameters in Database 153
5.6.2.2 Managing Global Parameters via the Adapter Global Parameter Management 153
5.6.2.3 Controlling Global Parameters in the X4 Control Center 153
5.6.2.4 Creating Global Parameters and Groups 153
5.6.2.5 Copying Global to Project Parameters 159
5.6.2.6 Using Global Parameters 161
5.6.3 Project Parameters 165
5.6.3.1 Managing Project Parameters via the Adapter Project Parameter Management 166
5.6.3.2 Controlling Project Parameters in the X4 Control Center 166
5.6.3.3 Creating Project Parameters 166
5.6.3.4 Renaming Project Parameters 172
5.6.3.5 Creating custom types 175
5.6.3.6 Reusing Global and System Parameters 181
5.7 XSL Help Functions 183
5.7.1 TimeHelper (deprecated) 184
5.7.1.1 Date/Time Methods 184
5.7.1.2 Example 190
5.7.2 WorkingTimeCalculator (deprecated) 194
5.7.2.1 Methods for Defining the Working Time 194
5.7.2.2 Working Time Calculating Functions 195

5.7.2.3 Example 198
5.7.3 checkXML 199
5.7.3.1 Method Description 199
5.7.3.2 Example 199
5.8 XSLT functions 200
5.9 XPath Functions 200
5.9.1 Node-set Functions 200
5.9.2 Boolean Functions 201
5.9.3 Number Functions 202
5.9.4 String Functions 202
5.10 Developer Utilities 204
5.11 URL Protocols 206
5.11.1 context: URL Protocol 206
5.11.2 xstore:// URL Protocol 207
5.11.3 x4db:/ URL Protocol 208
5.11.4 cryptstore:// URL Protocol 209
5.11.5 globalcontext: URL Protocol 211
5.12 Logging Technical Processes 212
5.12.1 Activating .log file 213
5.13 Using Save Points for resumable processes 214
5.14 Deploying Web Services via HTTP(S) 215
5.14.1 Editing the Web Service Configuration graphically 216
5.14.1.1 Calling the Webservice Configuration Editor 216
5.14.1.2 Interface of the Webservice Configuration Editor 216
5.14.1.3 Managing and Creating Constants, Variables or Remainders 218
5.14.1.4 Defining Safety Properties of the Node 218
5.14.2 Providing ReSTful Web Services 219
5.14.2.1 Deploying Processes as ReSTful HTTP Web Service 219
5.14.2.2 ReSTful Web Services: Input XML structure 221
5.14.2.3 ReSTful Web Services: Expected Output XML 222
5.14.3 Providing SOAP Web Services 223

5.14.3.1 Providing SOAP Web Services in RPC-Style 225
5.14.3.2 Providing SOAP Web Services in Document Mode 228
5.14.3.3 Providing SOAP Web Services with External WSDL Definition 231
5.14.3.4 Providing SOAP Web Services without WSDL Definition 233
5.14.3.5 SOAP Web Services: Input and Output XML 234
5.14.4 Providing File Web Services 237
5.14.4.1 Providing File Resources via HTTP 238
5.14.4.2 Using File Web Services with Variables 238
5.14.4.3 Using File Web Services with Remainder 239
5.14.4.4 Caching of Provided File Resources 240
5.15 Providing Web Services via Open API 240
5.15.1 Editing the Web Service Configuration graphically 241
5.15.1.1 Calling the Webservice Configuration Editor 241
5.15.1.2 Interface of the Webservice Configuration Editor 241
5.15.1.3 Managing and Creating Constants, Variables or Remainders 242
5.15.1.4 Defining Safety Properties of the Node 242
5.15.2 Providing ReSTful Web Services 243
5.15.3 Deploying a Technical Process ass OpenAPI ReST Service 243
5.15.3.1 Working with a Data Model 244
5.15.3.2 Request with a Data Model 245
5.15.3.3 Response with a Data Model 245
5.15.3.4 Setting the Authentication 246
5.15.3.5 Testing the ReST Service 246
5.15.3.6 Testing a ReST Service with Authentication 247
5.16 COBOL Copybook Converter Wizard 247
5.16.1 Requirements 248
5.16.2 Functions of the COBOL Copybook Wizard 248
5.17 Extracting the SessionID of a Web App 249
5.18 Read out groups of a user 249
5.19 Read out claims of a user 249
6 Debugging and Running Processes 250

6.1 Debug Technical Processes 250
6.1.1 Execute debug session without debugging 251
6.1.2 Terminate debug session 252
6.1.3 Attach debugger with server 252
6.2 Executing Processes 253
6.3 Status Values for Processes and Process Steps 253
7 Scheduler 255
7.1 Create scheduler 255
7.2 Execute processes scheduled 256
7.2.1 Create new scheduled execution 256
7.2.2 Server start 256
7.2.3 Cron 257
8 Process Monitoring 259
8.1 Configuring the Monitoring of Profiling Data in the Design View 259
8.2 Configuring the Monitoring of Profiling Data in the Source View 261
8.2.1 Example of .profiling file 261
9 Working with Adapter Projects 263
10 X4 Adapters Overview 264
10.1 About X4 Adapters 264
10.2 Big Data 264
10.2.1 Amazon S3 DynamoDB Connector 264
10.2.2 Apache Cassandra Connector 275
10.2.3 Google Cloud Storage BigTable Connector 283
10.2.4 Microsoft Azure Data Lake Store Connector 289
10.3 Cloud 296
10.3.1 Amazon S3 Simple Storage Connector 296
10.3.2 Amazon SimpleDB Connector 306
10.3.3 Dropbox Connector 313
10.3.4 Google Cloud Storage Objects Connector 316
10.3.5 Google Drive Connector 325
10.3.6 Microsoft Azure Blob Storage Connector 330

10.3.7 Microsoft Azure Table Storage Connector 337
10.4 Communication 342
10.4.1 AS/400 PCML 343
10.4.2 File System 344
10.4.2.1 Operation: Write File 347
10.4.2.2 Operation: Append File 348
10.4.2.3 Operation: Read File 350
10.4.2.4 Operation: Delete 351
10.4.2.5 Operation: Copy 351
10.4.2.6 Operation: Move 352
10.4.2.7 Operation: Exists 353
10.4.2.8 Operation: Create Folder 354
10.4.2.9 Operation List Folder 355
10.4.3 File System Bulk 356
10.4.3.1 Example for List 360
10.4.4 FTP 364
10.4.4.1 FTP Connector 364
10.4.4.2 FTP Connector (Advanced) 374
10.4.5 iCalendar 382
10.4.5.1 Operation: iCal to XML 383
10.4.5.2 Operation: XML to iCal 383
10.4.5.3 Operation: Create iXmap for E-Mail Adapter 387
10.4.6 Ping 388
10.4.7 SCP 389
10.4.8 TCP/IP Connector/Listener 392
10.4.8.1 TCP/IP Communication 392
10.4.9 Telnet Connector 393
10.5 Compression 395
10.5.1 gzip Converter 395
10.5.2 ZIP 396
10.5.2.1 Zip Converter 396

10.6 Databases 400
10.6.1 JDBC Connector 400
10.6.1.1 About the JDBC Connector 400
10.6.1.2 Statement: Direct SQL 404
10.6.1.3 StatementBatch: Direct SQL in Batch Mode 405
10.6.1.4 PreparedStatement: SQL with Values 406
10.6.1.5 PreparedStatementBatch: SQL with Values in Batch Mode 409
10.6.1.6 CallableStatement: Stored Procedures 411
10.6.1.7 CallableStatementBatch: Stored Procedures in Batch Mode 414
10.6.1.8 NamedPreparedStatement: SQL with Named Parameters 417
10.6.1.9 NamedPreparedStatementBatch: SQL with Named Parameters in Batch Mode 419
10.6.1.10 Passing XML Structures 421
10.6.1.11 Retrieving Database Metadata 422
10.6.2 JDBC Iterator 423
10.6.2.1 Statement: Direct SQL 426
10.6.2.2 StatementBatch: Direct SQL in Batch Mode 428
10.6.2.3 PreparedStatement: SQL with Values 429
10.6.2.4 PreparedStatementBatch: SQL with Values in Batch Mode 432
10.6.2.5 CallableStatement: Stored Procedures 434
10.6.2.6 CallableStatementBatch: Stored Procedures in Batch Mode 437
10.6.2.7 NamedPreparedStatement: SQL with Named Parameters 440
10.6.2.8 NamedPreparedStatementBatch: SQL with Named Parameters in Batch Mode 442
10.6.2.9 Retrieving Database Metadata 444
10.6.3 JDBC Metadata Explorer 445
10.6.3.1 Retrieving database metadata 447
10.6.3.2 Retrieving table metadata 448
10.6.3.3 Retrieving column metadata 450
10.6.3.4 Retrieving primary keys of a table 453
10.6.3.5 Retrieving foreign keys of a table 454
10.6.3.6 Retrieving referenced foreign keys 455

10.7 EDI 457
10.7.1 EDIFACT Adapter 457
10.7.2 Supported EDIFACT Message Types 465
10.8 E-Mail 470
10.8.1 IMAP 470
10.8.2 MIME Multipart 475
10.8.2.1 Operation FromBytesWithHeaders 476
10.8.2.2 Operation FromRestMetaXml 479
10.8.2.3 Operation ModifyRestMetaXml 484
10.8.2.4 Operation ToRestMetaXml 487
10.8.2.5 Operation ToBytesWithHeaders 489
10.8.2.6 Operation FromBytesWithBoundaryFromFirstLine 491
10.8.2.7 Operation ToBytesWithBoundaryInFirstLine 493
10.8.2.8 Operation FlattenXml 495
10.8.2.9 Operation PartFromX4Document 497
10.8.3 MIME Multipart Reader 498
10.8.3.1 Sample 498
10.8.3.2 GetDocument 499
10.8.3.3 Traverse 499
10.8.4 POP/SMTP 500
10.9 ERP 507
10.9.1 Abas ERP Connector 507
10.9.2 SAP Connector 513
10.9.2.1 IDoc structures 513
10.9.2.2 SAP Connector (JCo 2) 515
10.9.2.3 SAP Connector (JCo 3) 530
10.9.2.4 SAP Explorer (JCo 2) 542
10.9.2.5 SAP Explorer (JCo 3) 544
10.9.2.6 SAP Listener (JCo 2) 547
10.9.2.7 SAP Listener (JCo 3) 556
10.10 File Processing 566

10.10.1 ASCII to XML 566
10.10.1.1 ASCII to XML Converter 566
10.10.1.2 ASCII Writer 567
10.10.1.3 Text to Lines Converter 567
10.10.1.4 Text to Lines Converter from URL 570
10.10.2 Base64 Batch Decoder 572
10.10.3 Base64 Converter 573
10.10.4 Charset Changer 574
10.10.5 CSV 575
10.10.5.1 CSV Converter 575
10.10.5.2 CSV Converter from URL 580
10.10.5.3 XML to CSV Converter 586
10.10.6 Encoding Converter 589
10.10.7 Hex Converter 590
10.10.8 Image Processing 591
10.10.8.1 Image to Image Converter 591
10.10.8.2 Multipage TIFF Creator 592
10.10.8.3 TIFF Properties Reader 594
10.10.9 Java Properties File Reader 595
10.10.10 QR Code Generator 596
10.10.11 Regular Expression 598
10.10.12 Stream Splitter 599
10.10.13 Structured Text 600
10.10.13.1 Structured Text Converter 600
10.10.13.2 COBOL Copybook Reader 601
10.10.13.3 COBOL Copybook Writer 602
10.10.14 Text Replacement 603
10.10.15 Transliteration Adapter 604
10.11 Geo Location 605
10.11.1 Bing Maps 605
10.11.1.1 Operation: Geocode 607

10.11.1.2 Operation: ReverseGeocode 607
10.11.1.3 Operation: GetDirections 608
10.11.1.4 Operation: CreateBingURL 609
10.11.2 Geo Calculation 610
10.11.2.1 Operation: Line Conversion 614
10.11.2.2 Operation: LineLength 614
10.11.2.3 Operation: LineIntersection 615
10.11.2.4 Operation: BoundingBox 616
10.11.2.5 Operation: SegmentLineByPolygon 616
10.11.2.6 Operation: LineSimplificationPercent / LineSimplificationKilometer 618
10.11.3 Google Maps 619
10.11.3.1 Operation: Geocode 621
10.11.3.2 Operation: ReverseGeocode 622
10.11.3.3 Operation: GetDirections 622
10.11.3.4 Operation: CreateGoogleURL 623
10.12 Insurance 624
10.12.1 Audatex DV90 Converter 625
10.12.2 GDV 625
10.12.2.1 GDV Message Plausibility Checker 625
10.12.2.2 GDV Message Reader 626
10.12.2.3 GDV Message Writer 627
10.12.2.4 GDV-VU 627
10.12.3 SZF Reader 630
10.13 Internet of Things 631
10.13.1 IEC 61850 Client 631
10.13.2 Modbus TCP Connector 635
10.13.3 MQTT Publisher/Subscriber 638
10.13.3.1 MQTT Publisher 638
10.13.3.2 MQTT Subscriber 644
10.13.4 OPC Client Connector 650
10.13.5 OPC UA Publisher/Subscriber 653

10.13.5.1 OPC UA Publisher Adapter 653
10.13.5.2 OPC UA Subscriber Adapter 658
10.13.6 SIMATIC S7 Connector 666
10.14 Message Oriented Middleware 668
10.14.1 Apache Kafka 668
10.14.1.1 Apache Kafka Publisher 670
10.14.1.2 Apache Kafka Subscriber 675
10.14.2 AMQP Publisher/Subscriber 680
10.14.2.1 AMQP Publisher 680
10.14.2.2 AMQP Subscriber 684
10.14.3 JMS 687
10.14.4 Websphere MQ 693
10.14.4.1 IBM MQ Monitoring 693
10.14.4.2 MQ Series Transfer 695
10.14.4.3 Websphere MQ Connector 708
10.15 Office 718
10.15.1 Microsoft Excel 718
10.15.1.1 Microsoft Excel Reader 718
10.15.1.2 Microsoft Excel Writer 721
10.15.2 Microsoft Word Properties 730
10.15.3 Microsoft Word to Text Converter 732
10.15.4 Working Time Calculator 733
10.16 PDF Processing 737
10.16.1 PDF Data Processing 737
10.16.1.1 PDF Barcode Adder 737
10.16.1.2 PDF Form 740
10.16.1.3 PDF Metadata 747
10.16.1.4 PDF to Text Converter 750
10.16.2 PDF Image Processing 752
10.16.2.1 PDF Image Adder 752
10.16.2.2 PDF Multipage TIFF Adder 754

10.16.2.3 PDF to Image Converter ... 756
10.16.2.4 PDF to TIFF Converter ... 758
10.16.2.5 TIFF to PDF Converter ... 760
10.16.3 PDF Split and Merge ... 762
10.16.3.1 PDF Merger ... 762
10.16.3.2 PDF Page Counter ... 763
10.16.3.3 PDF Recombiner ... 763
10.16.3.4 PDF Splitter Iterator ... 765
10.17 Security ... 768
10.17.1 Antivirus ... 768
10.17.1.1 Setup ClamAV ... 770
10.17.1.2 Updating the virus database ... 770
10.17.2 BCrypt Calculator ... 770
10.17.3 Checksum Creator ... 772
10.17.4 Digest Calculator ... 773
10.17.5 JSON WebToken Calculator ... 774
10.17.6 LDAP Connector ... 778
10.17.6.1 search command ... 780
10.17.6.2 add command ... 781
10.17.6.3 modify command ... 781
10.17.6.4 modifyDN (RDN) command ... 782
10.17.6.5 delete command ... 783
10.17.6.6 compare command ... 783
10.17.7 Password Generator ... 784
10.17.8 Random Number Generator ... 786
10.17.9 RSA Encryption ... 787
10.17.10 SAML Calculator ... 788
10.17.11 SSH Connector ... 792
10.17.12 TOTP One-Time Password Generator ... 794
10.17.13 UUID Generator ... 796
10.17.14 XML Encryption ... 798

10.17.14.1 XML Encryption (Keystore) 798
10.17.14.2 XML Encryption (Password) 801
10.17.15 XML Signature 803
10.17.16 AES Encryption 805
10.18 System Administration 811
10.18.1 JMeter Test Execution 811
10.18.2 JMX MBean 817
10.18.2.1 Betrieb: MBeans finden 818
10.18.2.2 Operation: Execute Operations 819
10.18.3 JMX Number 822
10.18.4 JMX Switch 823
10.18.5 SNMP Management 825
10.18.6 Windows Event Reader 829
10.18.7 Windows Services Explorer 831
10.19 Tools 832
10.19.1 WebSMS 832
10.20 Web Services 833
10.20.1 JSON 833
10.20.1.1 JSON Converter 2.0 833
10.20.1.2 JSON Validation 837
10.20.2 OData Connector 838
10.20.3 URL Encoding Converter 849
10.20.4 URL Reader 851
10.20.5 Webservice 852
10.20.5.1 HTTP(S) Connector 852
10.20.5.2 Webservice (SOAP) Connector 865
10.20.5.3 Webservice Security 868
10.20.6 X4 Streaming Support 871
10.20.7 XOP (MTOM) 873
10.20.7.1 Operation CleanupXml 874
10.20.7.2 Operation MakeSimplepart 875

10.20.7.3 Operation MakeMultipart 876
10.20.7.4 Operation UnpackMultipart 880
10.20.8 YAML Converter 884
10.21 X4 API 886
10.21.1 Deployments 886
10.21.1.1 Examples 887
10.21.2 Global Parameter Management 891
10.21.3 Keycloak Management 893
10.21.3.1 Users 895
10.21.3.2 Groups 912
10.21.3.3 Roles 927
10.21.4 Process Monitoring 953
10.21.4.1 Operation: Find Process Execution Detail 954
10.21.4.2 Operation: List Process Executions 955
10.21.4.3 Operation: List Process Execution Steps Detail 964
10.21.5 Process Monitoring Management 965
10.21.5.1 Operation List Profilings 966
10.21.5.2 Operation Get Profiling by Id 967
10.21.5.3 Operation Update Profiling 968
10.21.6 Process Runtime Management 970
10.21.6.1 Operation: List Running Processes 971
10.21.6.2 Operation: Stop Process 971
10.21.6.3 Operation: Terminate Process 972
10.21.7 Project Parameter Management 972
10.21.7.1 Operation: Get Project Parameter Ids 974
10.21.7.2 Operation: Get Project Parameter Structure 975
10.21.7.3 Operation: List Project Parameters 977
10.21.7.4 Operation: Patch Project Parameters 978
10.21.8 Savepoint Management 979
10.21.8.1 Operation: List Savepoints 980
10.21.8.2 Operation: Delete Savepoint 983

10.21.8.3 Operation: Resume Savepoint 984
10.21.9 Schedule Management 985
10.21.9.1 Operation: List Schedules 986
10.21.9.2 Operation: Get Schedule by Id 986
10.21.9.3 Operation: Update Schedule 988
10.21.9.4 Operation: Jobs Information 989
10.21.10 System Configuration Management 991
10.21.10.1 Operation: List Configurations 992
10.21.10.2 Operation: Get Configuration by configurationName 992
10.21.11 Third Party License Summary 995
10.21.11.1 Operation: Get Summary 995
10.21.12 Web App Configuration Management 996
10.21.12.1 Operation: Get Configuration by Id 997
10.21.12.2 Operation: List WebApp Configurations 998
10.21.12.3 Operation: Update Configuration 999
10.21.13 Web App Theme Management 1000
10.21.13.1 Operation: Update Theme 1001
10.21.13.2 Operation: Get Theme by Id 1003
10.21.13.3 Operation: List WebApp Themes 1005
10.22 X4 Tools 1006
10.22.1 Case Management Documents 1006
10.22.2 Case Management 1009
10.22.3 Case Management Statistics 1012
10.22.4 Commandline 1015
10.22.5 Document Comparison 1018
10.22.6 Document Properties Changer 1020
10.22.7 Empty Output Adapter 1021
10.22.8 Global Context Manipulation 1022
10.22.9 Keycloak Token 1024
10.22.10 Log4J Logging 1025
10.22.11 Loop Adapter 1026

10.22.12 Parallel Process Starter 1027
10.22.13 Process Starter 1030
10.22.14 Projects 1032
10.22.15 Project Access 1034
10.22.15.1 Operation: ReadFile 1035
10.22.15.2 Operation: WriteFile 1036
10.22.15.3 Operation: AppendFile 1037
10.22.15.4 Operation: Exists 1039
10.22.15.5 Operation: CreateDirectory 1040
10.22.15.6 Operation: Delete 1040
10.22.15.7 Operation: ListDirectory 1041
10.22.15.8 Operation: Move 1042
10.22.15.9 Operation: Copy 1043
10.22.16 System Status 1044
10.22.16.1 Operation: Threads Information 1045
10.22.16.2 Operation: JVM Parameters 1046
10.22.16.3 Operation: Processes Running 1047
10.22.16.4 Operation: User Sessions Memory 1048
10.22.16.5 Operation: Apps Status 1049
10.22.17 Task Management 1050
10.22.18 User Identity Adapter 1053
10.22.18.1 Output XML 1055
10.22.18.2 XSD 1056
10.22.19 User Session Adapter 1057
10.22.20 Web App Translation 1058
10.22.21 X4 Management 1059
10.22.22 X4 Server Config Reader 1060
10.22.23 X4 Variable Collector 1061
10.22.23.1 Operation Collect 1062
10.22.23.2 Operation Get 1063
10.22.23.3 Operation GetAndRemove 1064

10.23 XML Tools 1064
10.23.1 Inline Content Expander 1064
10.23.2 Record Counter 1067
10.23.3 XHTML Form Extractor 1068
10.23.3.1 Operation: ExtractFields 1072
10.23.3.2 Operation: FillInFields 1072
10.23.4 XML Checker 1073
10.23.5 XML Element Wrapper 1073
10.23.6 XML Formatter 1074
10.23.7 XML Iterator 1075
10.23.8 XML Text Extractor 1076
10.23.9 XML Validation 1078
10.23.10 XPath Iterator 1080
10.24 Deprecated adapters 1081
10.24.1 Atlassian Jira Connector 1081
10.24.1.1 Get Issue 1084
10.24.1.2 Get all Issues 1091
10.24.1.3 Create Issues 1101
10.24.1.4 Edit Issues 1102
10.24.1.5 Add Attachment 1103
10.24.1.6 Remove Attachment 1105
10.24.1.7 Get all Comments 1105
10.24.1.8 Get Comment 1108
10.24.1.9 Add Comment 1110
10.24.1.10 Edit Comment 1111
10.24.1.11 Search 1113
10.24.1.12 Execute Transition 1114
10.24.1.13 Get Transition 1115
10.24.2 Atlassian Jira Connector ReST API 1116
10.24.3 Global Parameter Initializer (previously: Custom Placeholder Initializer) (deprecated) 1118

10.24.4 MongoDB Connector 1120
10.24.4.1 About the adapter 1120
10.24.4.2 Configuring connection 1124
10.24.4.3 Coding of a document structure 1125
10.24.4.4 Supported data types 1126
10.24.4.5 Input 1126
10.24.4.6 Samples 1131
10.24.5 Repository Explorer (deprecated) 1136
10.24.6 JSON Converter (deprecated) 1139
10.24.7 ZIP Compression (Deprecated) 1140
10.24.8 ZIP Folder Compression (Deprecated) 1141
10.24.9 ZIP in Memory Compression (Deprecated) 1142

1 Introduction

1.1 What is an ESB Project?

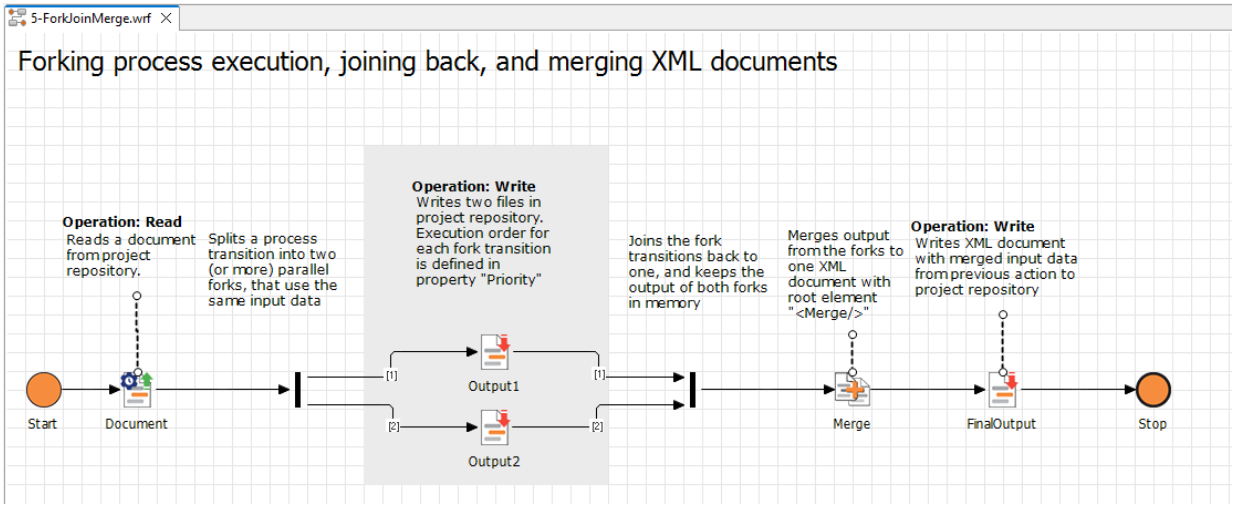
An ESB (Enterprise Service Bus) Project is a project type within X4 BPMS that allows you to graphically model and execute Technical Processes. ESB Projects contain a predefined folder structure in which any number of Technical Processes (*.wrf) can be created.

1.2 How can I use ESB Projects?

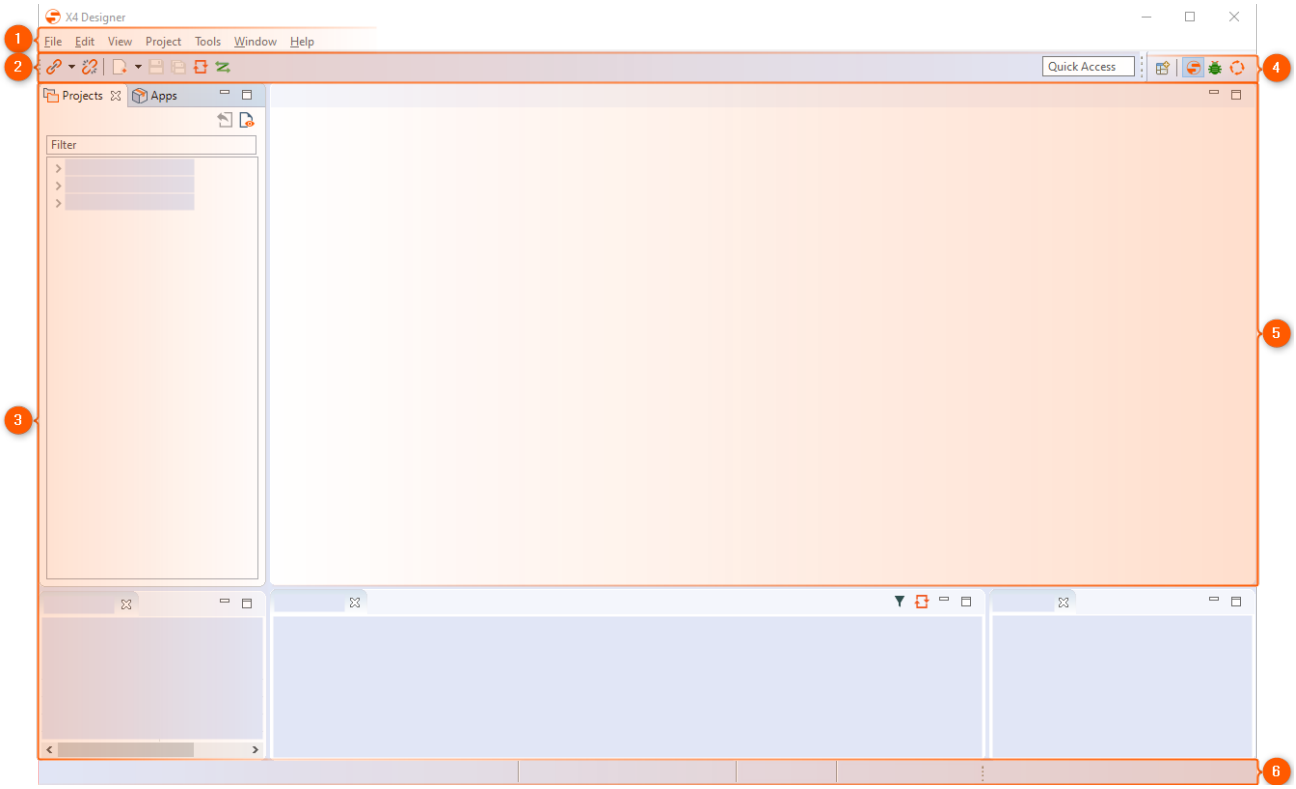
Within ESB Projects you can create technical processes and model them graphically. All process modules are simply dragged and dropped into the drawing area. There, after parameterization, they can be tested directly via the integrated debugging tool or executed on the X4 server on a time- and event-controlled basis.

The application possibilities of ESB Projects are very diverse. Examples of use include electronic data exchange between systems via the available adapters or the transformation of data via mappings.

Forking and Merging in a Technical Process



2 Interface



1	**Menu bar**	The menu bar allows to open various functions and editors of the X4 Designer.
2	**Toolbar**	The toolbar offers different functions and display options. Some symbols are only active, if the corresponding function is available for the selected or opened element, see also Toolbar.
3	**Views**	Views can be arranged individually via drag-and-drop.
4	**Perspectives**	Switch between X4 Designer and BPMN Designer perspective.
5	**Editors and workspace**	Processes and files are opened here for editing in tabs.
6	**Status bar**	The status bar displays the connected server, the logged-in user and the version.

2.1 Toolbar

Various functions and display options can be called up via the toolbar. Some icons are only active if the relevant function is available for the selected or opened element.

	Connect	Connect to the server
	Disconnect	Disconnect from the server
	New	Add new element The insert option varies depending on where you are in the tree structure in the Repository Navigator.
	Save	Save changes in the currently opened window
	Save all	Save all changes
	Repository Refresh	Perform server-side update
	Attach/Detach Debug	Attach debugger to server instance / Detach debugger from server instance
	Start Debugging	Start Debugger
	Undo	Undo last editing step
	Redo	Restore undone step
	Cut	Cut selection
	Copy	Copy selection
	Paste	Paste selection
	Start Debugging	Call up options for process execution • `Start Debugging`: debug process (show and fix errors) • `Start Without Debugging`: Execute process directly • `Debug/Run Options`: Open Run/Debug settings in X4 Designer
	Suspend	Stop debugging process
	Resume	Continue debugging process
	Stop Debugging	Terminate debugging

	Step Into	Execute process step (display steps of the subprocess)
	Step Over	Execute process step (without displaying steps of the subprocess)
	Step Return	Jump out of subprocess
	Toggle Breakpoint	Set or remove breakpoint for marked element
	Clear all Breakpoints	Delete all breakpoints in this process
	Quick Watch	Open Quick Watch view to view placeholders, dynamic parameters and context variables with their values during debugging.
	Output	Open Output view to view the result of each process step during debugging.
	Zoom Out	Zoom out view
	Zoom	Select zoom level
	Zoom In	Zoom view
	Snap to grid	Display grid and align elements to grid
	Show Process Quality Index	Display Process Quality Index

2.2 Context Menu

Context menu for process components

When you right-click a process component in the X4 Designer's editor area, you have access to different features – depending on the component type and the:

Add Component	Add new process component

Layout	Align selected process components • `Align Horizontal`: Align the process components horizontally • `Align Vertical`: Align the process components vertically • `Distribute Horizontal`: Distribute the process components horizontally • `Distribute Vertical`: Distribute the process components vertically
Select All	Select all process components
Copy Process Diagram to Clipboard	Copy the process diagram
Edit Label...	Edit the process component's label
Undo ...	Undo the last editor action
Redo ...	Redo a previously undone editor action
Copy	Copy a selected process action
Cut	Cut selected process action
Set ... document	Replace a marked process component with another repository element
Edit ...	Edit a selected process component (module components and documents only)
Toggle Breakpoint	Set a debug breakpoint for the selected process action
Delete	Delete a selected process component from the process diagram
Validate	Check the subprocess' validity (subprocesses only)
Preview	Open process component in a separate editor tab (Module components and documents only)
Show input schema	Display input schema assigned to the sub-process, if available
Show output schema	Display output schema assigned to the sub-process, if available
Select in Repository Navigator	Show process component in the **Repository Navigator**
View Call Hierarchy	Display the processes using the selected process component
Refresh	Reload a selected process component from the repository (Module components and documents only)
Bring to Front	Bring a selected symbol to the foreground of the diagram

Bring Forward	Bring a selected symbol one level up towards foreground
Send Backward	Bring a selected symbol one level down towards background
Send to Back	Bring a selected symbol to the background of the diagram
Properties	Display process component properties in the Properties view
Schema Declaration ...	Specify schema declaration for the process

Context menu in the Projects view

You can open the context menu by right-clicking on an element within the **Projects** view.

New	Create new elements (processes, directories, adapters, etc.) in the repository
Open / Open with	Open the selected element (process, folder, adapter etc.) with the default editor or a specific editor
Save As Template	Saves the selected element as template
Copy full path to Clipboard undCopy XStore URL to Clipboard	Copy the path of the selected element to the clipboard (as relative folder path or as `xstore://` URL)
Cut, Copy and Paste	Cut, copy, or pastes the selected element(s)
Delete	Delete selected element from the repository (irreversible)
Refactor/Rename	Rename an element (and change references, if desired)
Exclude from Project	Exclude the resource from the project • The file will remain in the file system, but is no longer part of the project and is grayed out. • The usual operations such as opening, editing, copying, deleting, etc. of excluded files are disabled.
Include to Project	Include an excluded resource to the project
Repair References	Repair incorrect repository resource paths in process diagrams
Validate	Check the selected process's validity
View Call Hierarchy	Display the processes that use the selected elements
Start Without Debugging	Run process directly
Import	Import resource to the repository
Export	Export project or resource
Clean-Up Project	Select and delete unused files from the repository

2.3 Perspectives

A perspective is a defined arrangement of views and the editor area. The arrangement of the views can be defined by drag-and-drop.

X4 Designer	The X4 Designer perspective is the default perspective.
Debug	The Debug perspective is an optimized perspective to debug business and technical processes.
BPMN Designer	The BPMN Designer perspective is an optimized perspective to model business processes.

2.3.1 Switching perspectives

In X4 Designer, you can switch between perspectives. The arrangement of the views is kept.

- To change the perspective, click on the icon of the desired perspective in the upper right corner.

2.3.2 Reseting perspectives

1. To reset a perspective, switch to the perspective.
2. Select **Window > Reset Window Layout** in the menu bar.

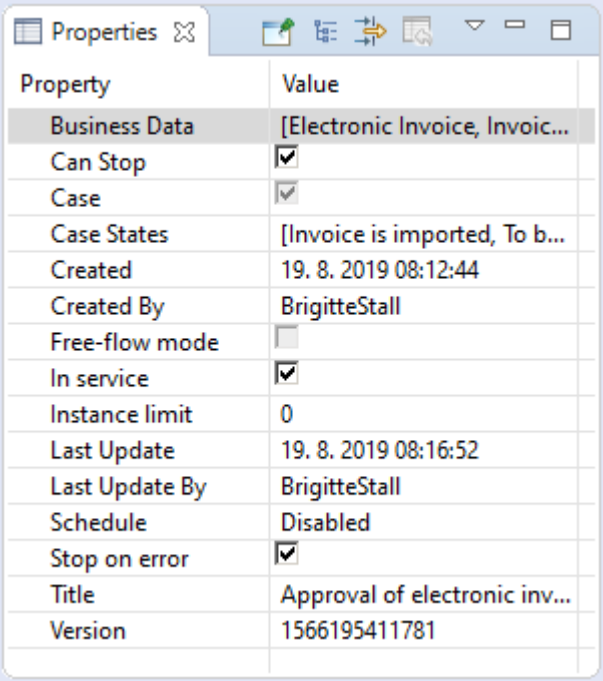
2.4 Views

Projects	The **Projects** view displays a tree structure of the connected repository with various elements such as projects, folders, and files. The context menu can be opened by right-clicking on an element in the **Projects** view. The available options change depending on where in the tree structure the context menu is called. The **Filter** text field above the project tree can be used to search for files within the connected repository. Placeholders like * or ? are not possible. The following actions can be performed using various buttons: • *Jump from active Editor to Repository Navigator*: Highlight opened file in the editor within the **Projects** view. • *Show or hide all files*: Show and hide contents of the file system. Files excluded from the project are grayed out.
Apps	The **Apps** view displays all X4 Apps that are used to bundle projects.

Properties

In the **Properties** view, the properties of selected process modules and of opened processes can be displayed and edited:

- The left column displays the property name, the right column contains the corresponding values.
- A title and a description can be stored for each process. These are displayed in the generated documentation.
- To edit properties of an opened process, click on an empty area on the workplace.



The most common properties are displayed by default, however, additional settings can be made using the **Properties** view toolbar:

- *Pin to Selection*: Pin displayed properties in the **Properties** view
- *Show Categories*: Show properties sorted by categories
- *Show Advanced Properties*: Show advanced properties
- *Restore Default Value*: Restore to default value
- *View Menu*: Open additional options

Run/Debug Console

If a process is checked for errors or simulated, debug information is displayed in the **Run/Debug Console** view for each executed process step. Error messages are highlighted in red, successful status messages are green.

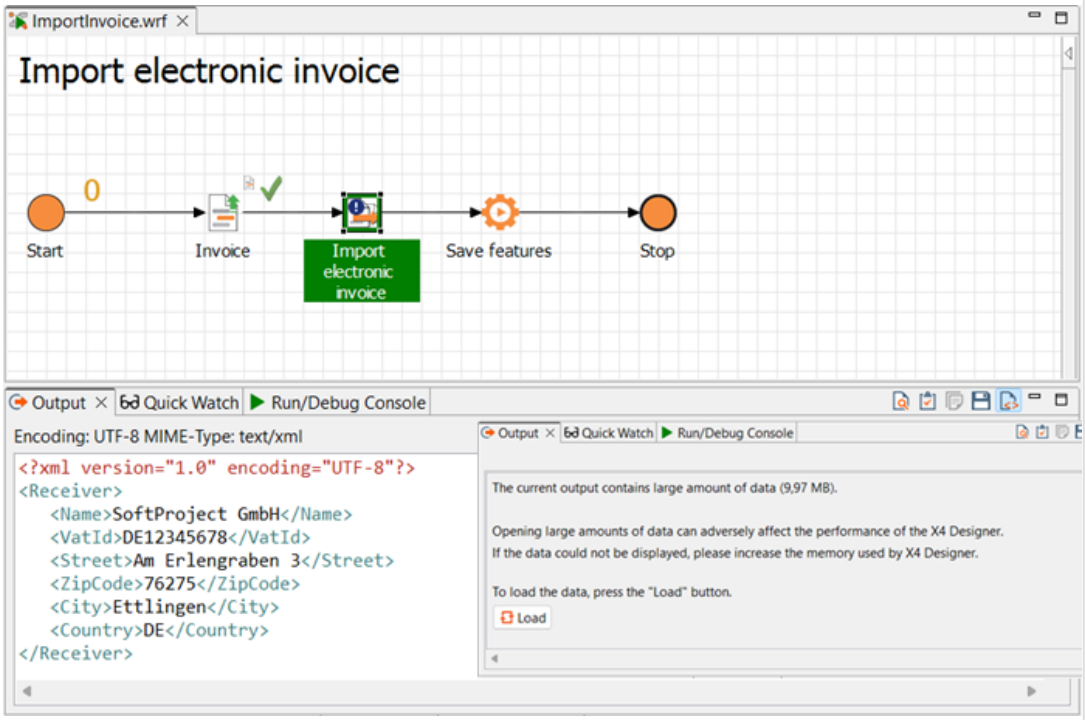
When a process step outputs an intermediate result, it is stored in a temporary file during debugging, which you can view using the corresponding reference (e.g., `file://1376054979039/`).

The **Run/Debug Console** view can be accessed from the **View** > **Run/Debug Console** menu.

Output

In the **Output** view, you can see the results of each debug step.

You can access the **Output** view during debugging from the **Debug** > **Output** menu or using the **Output** icon in the toolbar.

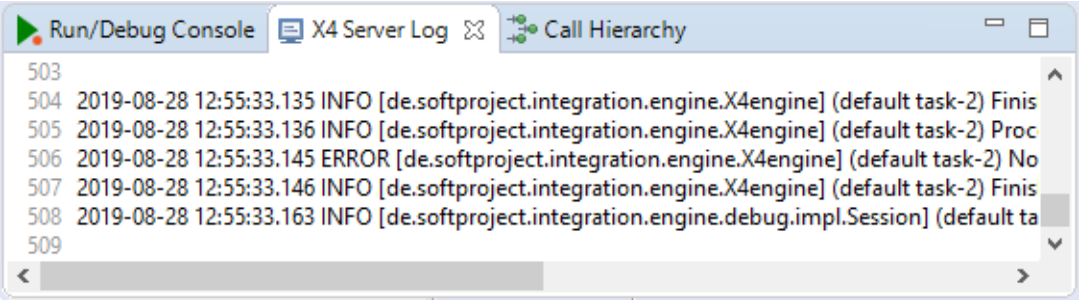
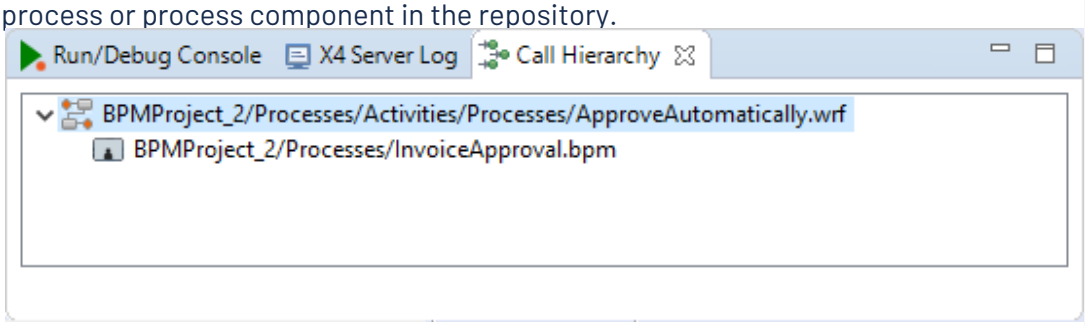


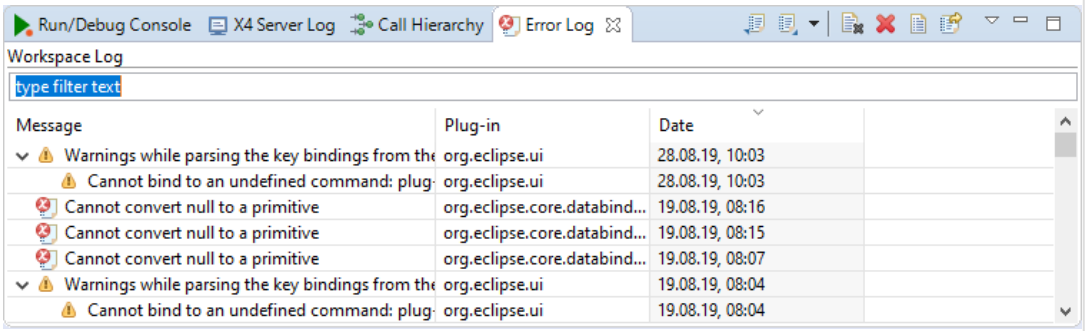
If the output is larger than 1 MB, a warning is displayed. You can download the data using the **Load** button.
Large amounts of data can affect the performance of the X4 Designer.

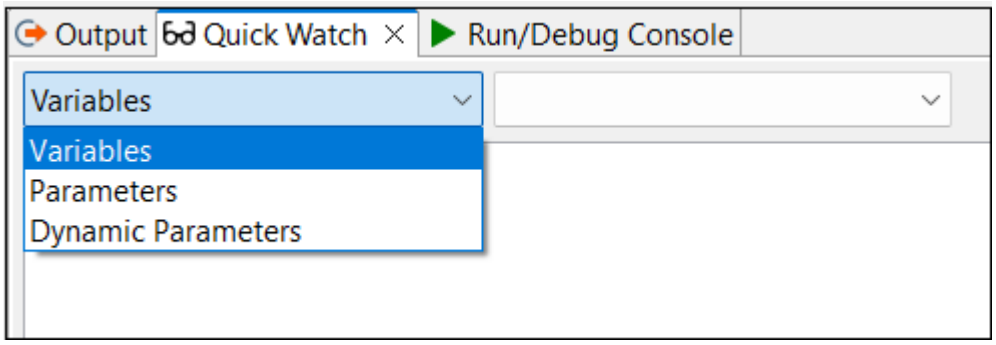
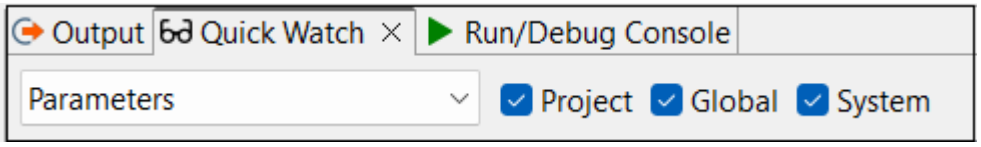
You can also perform the following actions from the toolbar:

- *Find content:* Find data in the **Output** view
- *Copy all content*: Copy all data from the **Output** view to the clipboard
- Copy selected content: Copy selected data from the **Output** view to the clipboard
- *Save content*: Save the data from the **Output** view
- *Format content*: Format the data from the **Output** view

Binary data is displayed as hexadecimal data in the **Output** view. Images and text-based contents are shown as image or text. The output of binary data as hexadecimal data does not affect the performance of the X4 Designer.

X4 Server Log	The **X4 Server Log** view allows to track the server status and especially errors in the **X4 Server Log** view from within the X4 Designer.  The **X4 Server Log** view can be accessed via the **View > X4 Server Log** menu.
Call Hierarchy	The **Call Hierarchy** view allows to display all repository elements using a certain process or process component in the repository.  Right-click on the desired element in the **Projects** view to bring up a context menu. **View Call Hierarchy** then displays a list of all repository items that contain at least one reference to the selected item.

Error Log	The **Error Log** view allows to view errors that occurred in the X4 Designer.  The following additional actions can be performed via the toolbar: • *Export Log*: Export error log and save it as `.log` file • *Import Log*: Import error log (`.log`) • *Clear Log Viewer*: Clear error log • *Delete Log*: Delete error log permanently • *Open Log*: Open log file (`.log`) in X4 Designer • *Restore Log*: Restore error log • *View Menu*: Open additional options The **Error Log** view can be opened via **View** > **Error Log**.
Problems	In the **Problems** view, all errors and warnings are listed. The filter can be used to customize the displayed list.
Debug Sessions	The **Debug Sessions** view lists all active debug sessions. Double-click the desired process in the view to switch to the debug session. Active debug sessions are marked in gray. New debug sessions connected to the server are highlighted in orange.
Breakpoints	In the **Breakpoints** view, all set breakpoints are listed. In the view, breakpoints can be activated, deactivated and deleted. Additionally, all breakpoints can be skipped without deleting the breakpoints.

Quick Watch	In the **Quick Watch** view, variables, parameters, and dynamic parameters can be viewed separately with their values during debugging. To do so, choose the corresponding entry from the drop-down list:  Selected variables are updated after each debug step. When you select the **Parameters** entry, you can use the **Project, Global** and **System** checkboxes to specify which type of parameters you would like to display:  ⚠ If the result of the debug step contains content greater than 1 MB, a warning is displayed and the result can be downloaded via a button. Large amounts of data can affect the performance of the X4 Designer. ℹ The **Quick Watch** view can be accessed during debugging via the **Debug** > **Quick Watch** menu or via the **Quick Watch** icon in the toolbar.
Outline	In the **Outline** view, the complete process diagram is displayed as an image. By clicking on the individual areas of the illustration, it is easy to navigate through the process, even if it is larger than the size of the Designer window. ℹ The **Outline** view can be opened via the **View > Outline** menu.
Base64	Base64 encode or decode strings
Digest	Generate hash value for strings
Regex	Apply regular expression (*Regex*) to a string to test the expression or to directly replace the string according to a pattern
URLCodec	URL encode or decode strings

2.4.1 Open views

> ⚠ No other views can be opened in the **BPMN Designer** perspective, all relevant views are open by default.

- Select the desired view from the **View** menu in the menu bar.

2.5 Editors

In X4 Designer, processes and files are opened in tabs for viewing and editing. The used editor depends on the opened file type.

Editor	Usage	Description
BPMN Editor	.bpm .pld	With the BPMN Editor, you can model business processes according to the notation standard BPMN 2.0.
Database Editor	.stm	With the Database Editor, you can define SQL queries for database connection modules to retrieve data from a relational database.
JSON Editor	.json	With the JSON Editor, you can edit JSON documents directly in the X4 Designer.
Mapping Editor	.xsl .rep	With the Mapping Editor, you can create and edit XSL mappings in a graphical view.
Parameter Editor	.param	With the Parameter Editor, you can create project parameters and project parameter groups to parameterize process components.
Process Editor	.wrf	With the Process Editor, you can graphically model Technical Processes.
XML Editor	.xml .xhtml	With the XML Editor, you can create and edit well-formed XML documents.

XSD Editor	.xsd	With the XSD Editor, you can create an XML Schema Definition for your XML documents.

2.5.1 BPMN Editor

The BPMN Editor allows you to model Business Processes in compliance to the notation standard BPMN 2.0. The editor is similar to the Process Editor within X4 Designer regarding layout and functionality, but differs regarding the process components and symbols in the process diagram. The BPMN Editor allows you to link your modeled Business Processes directly to Technical Processes. Without switching the development environment, you can model your Business Processes and Technical Processes, and test and execute them on the X4 Server.

Since BPMN processes can contain non-automatable actions, e. g. human workflows, BPMN processes aren't necessarily executable by the X4 Server. BPMN processes may contain technical processes as subprocess tasks, and can be executed.

> The f*ree-flow* mode additionally allows to use the BPMN Editor as plain BPMN drawing tool with an extended BPMN palette where the run/debug functionality is disabled.

2.5.2 Database Editor

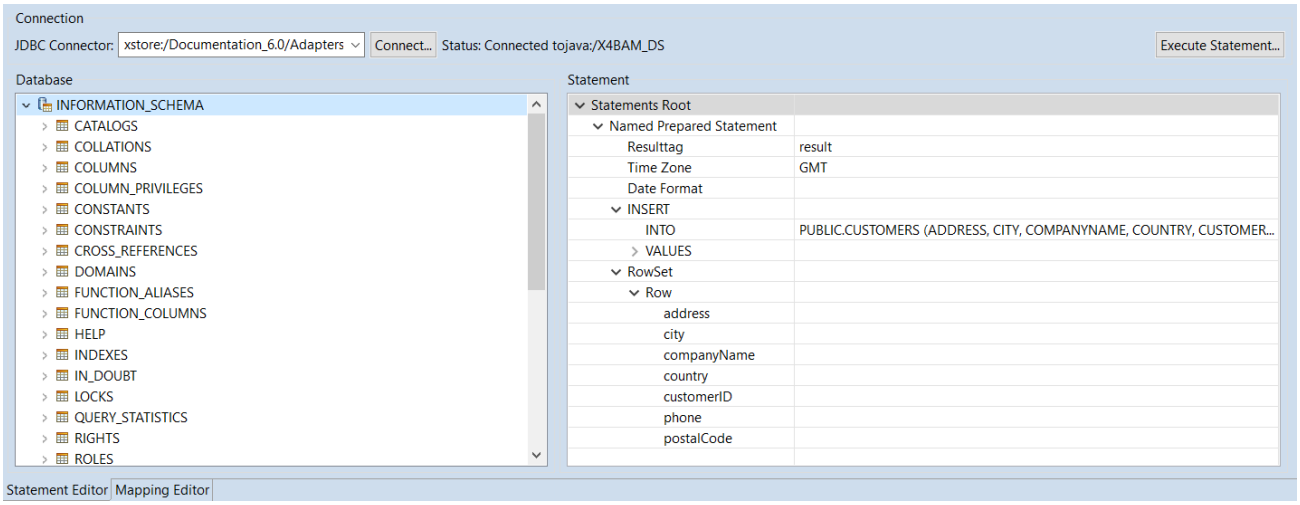
The Database Editor allows to define SQL queries for database connection components in order to retrieve data from a relational database. The logical separation of the process components SQL statement and database connection allows these components to be used flexibly and multiple times within the X4 Designer.

> To open the Database Editor create a new SQL statement file (`.stm`) or open an already existing SQL statement file.

The Database Editor's structure

The Database Editor provides two different tools:

- **Statement Editor**: Allows to connect to a database and displays the structure of the connected database. This structure can be used to create an SQL statement for the database.
- **Mapping Editor**: Allows to define a mapping for elements of an input XML document to the named parameters of the statement previously created with the Statement Editor. This mapping is subsequently used to create an XSL mapping, which generates a dynamically created input for the JDBC Connector using a suitable input XML document.



2.5.2.1 Statement Editor

The Statement Editor is divided into the following areas:

Connection area	Within this area a connection to a database can be established by means of a JDBC Connector. • **JDBC Connector**: xstore URL to a JDBC Connector • **Connect**: Connect to the database The connection status is displayed under **Status**. • **Execute Statement**: Execute sql statement
Database area	The database's structure is displayed here after establishing the connection. The following icons are used to visualize the structure: • Database schema • Database table • Primary key • Primary key/ Foreign key • Foreign key

Statement area

Here, different statements can be created for the connected database via the context menu:

- **New Named Prepared Statement**: Create a new Named Prepared Statement in `Statements Root`
- **Edit Named Prepared Statement**: Edit `Named Prepared Statement`
- **New RowSet**: Create new `RowSet`.
  Only one `RowSet` can exist within a Named Prepared Statements. It is generated automatically when creating the Named Prepared Statement.
  A new `RowSet` can only be created, if the existing one has been deleted.
- **New Row**: Add a new row within a `RowSet`
- **New Named Parameter Value**: Add a new Named Paramerter value within a row
- **Edit Named Parameter**: Edit Named Parameter
- **New SELECT Statement**: Create new `SELECT` statement
- **Edit SELECT**: Edit `SELECT` instruction
- **Edit FROM**: Edit `FROM` within a `SELECT` statement
- **New INSERT Statement**: Create new `INSERT` statement
- **Edit INTO**: Edit `INTO` instruction
- **New VALUE Entry**: Create new `VALUE` entry
- **Edit VALUE**: Edit `VALUE`
- **New UPDATE Statement**: Create new `UPDATE` statement
- **Edit UPDATE**: Edit `UPDATE` instruction
- **New SET Entry**: Create new `SET` entry
- **Edit SET Entry**: Edit `SET` entry
- **New DELETE Statement**: Create new `DELETE` statement
- **Edit FROM**: Edit `FROM` instruction
- **Edit WHERE**: Edit `WHERE` condition
- **New Column**: Add new column within an `INSERT` or `SELECT` instruction
- **New Table**: Insert new table within an `UPDATE` or `SELECT` instruction
- **Delete**: Delete element

> For further information on how to create and edit statements, see section Creating Statements for a Connected Database.

Drag&Drop actions

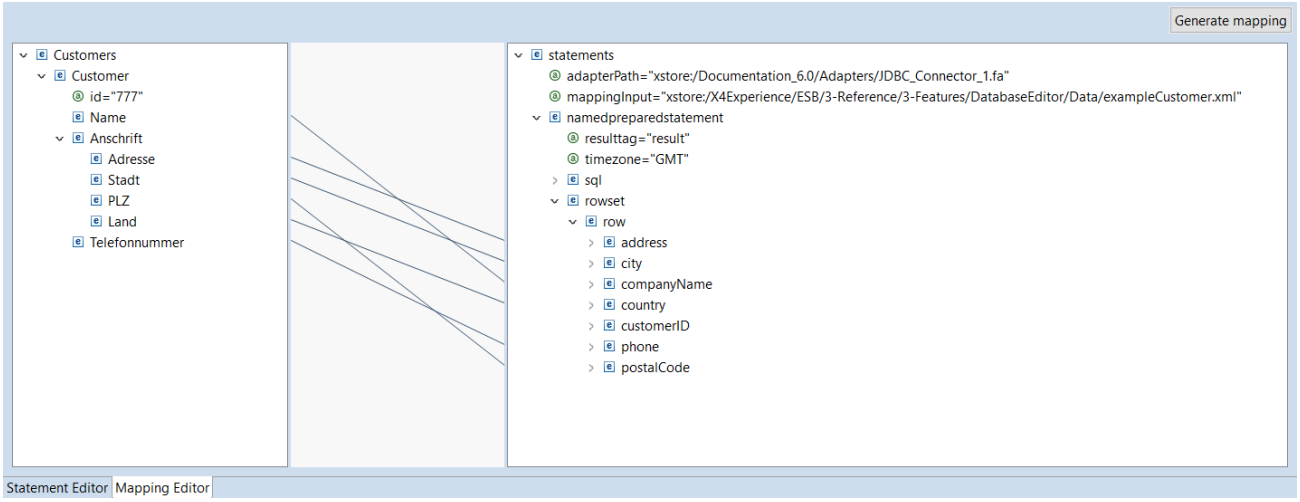
The following Drag&Drop actions can be performed within the Statement area:

SQL statement	Database element	Statement element	Action
SELECT	Column	SELECT	Add the column to be selected. If the table corresponding to the added column is not already entered within FROM, it will be entered automatically.
	Table	FROM	Add the table from which is to be selected.
UPDATE	Column	SET Entry	Set the column as entry. If the table corresponding to the added column is not already entered within UPDATE, it will be entered automatically.
	Table	UPDATE	Set the table to be updated.
INSERT	Table	INTO	Set the table into which is to be written. If there is already a table, it will be replaced. Already set columns will be deleted thereby.
	Column	INTO	Set the column into which is to be written. This is only possible, if the column originates from the set table.
DELETE	Table	FROM	Set the table to be deleted. If there is already a table, it will be replaced.

2.5.2.2 Mapping Editor

The Mapping Editor allows to define a mapping for elements of an input XML document to the named parameters of the statement previously created with the Statement Editor. This mapping is subsequently used to create an XSL mapping, which generates a dynamically created input for the JDBC Connector using a suitable input XML document.

The Mapping Editor is divided into the following areas:



Source area	The Source area is on the Mapping Editor's left side. The input XML file can be displayed here and assigned by drag & drop. The context menu provides the following options: • **Collapse**: Collapse a node's child nodes • **Expand**: Expand the child nodes of the selected node • **Remove Input**: Remove assigned input XML document
Linking area	The Linking area is located in the Mapping Editor's middle column. It visualizes the `xsl:value-of` connections between an element in the source document and a Named Parameter in the statement as line.
Statements area	The Statements area is located on the Mapping Editor's right side. The complete structure of the statement file is displayed here. Moreover, elements of an input XML file can be mapped to the child elements of a `Row` element.

2.5.3 JSON Editor

The JSON Editor allows JSON documents to be edited directly within the X4 Designer.

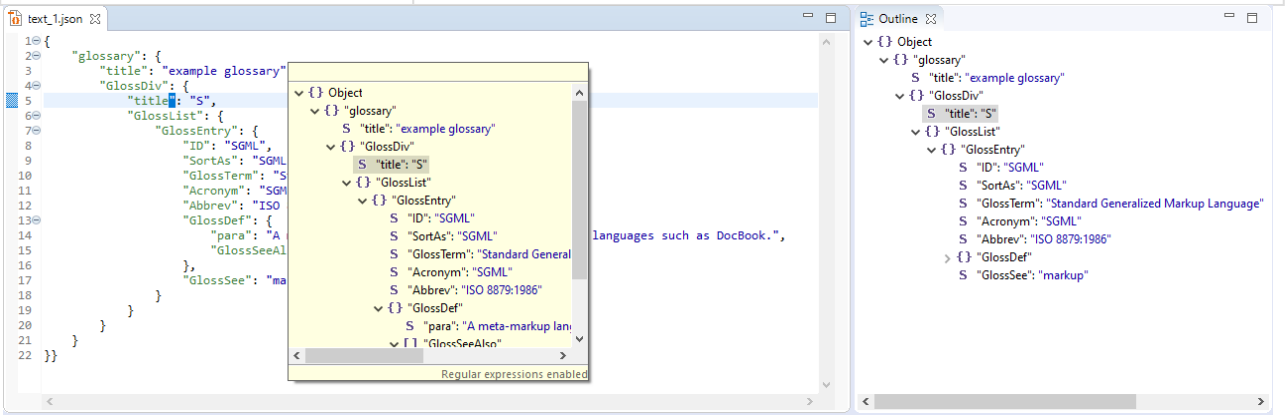
> ⓘ To open the JSON Editor, either create a new JSON Document or open an already existing document.

The Outline view displays the structure of the opened JSON document. Moreover, you can also use the context menu **Quick Outline** or the shortcut `CTRL+O` to open the Quick Outline view and to search for keys and values within the JSON document.

Shortcuts

`CTRL+O`	Open the Quick Outline view

CTRL+Shift+F	Format the JSON code



The formatting of the JSON code and the colors for syntax highlighting can be changed under **Tools > Options > JSON Editor**. For more information, see Configuring the JSON Editor.

2.5.4 Mapping Editor

The Mapping Editor allows to create and edit XSL mappings in a graphical interface. Using XSLT, you can transform an XML data structure into another XML data structure or into an XSL-FO document. The Mapping Editor supports you with a graphical interface and with an automatic stylesheet and XPath validation when creating or editing XSL mappings.

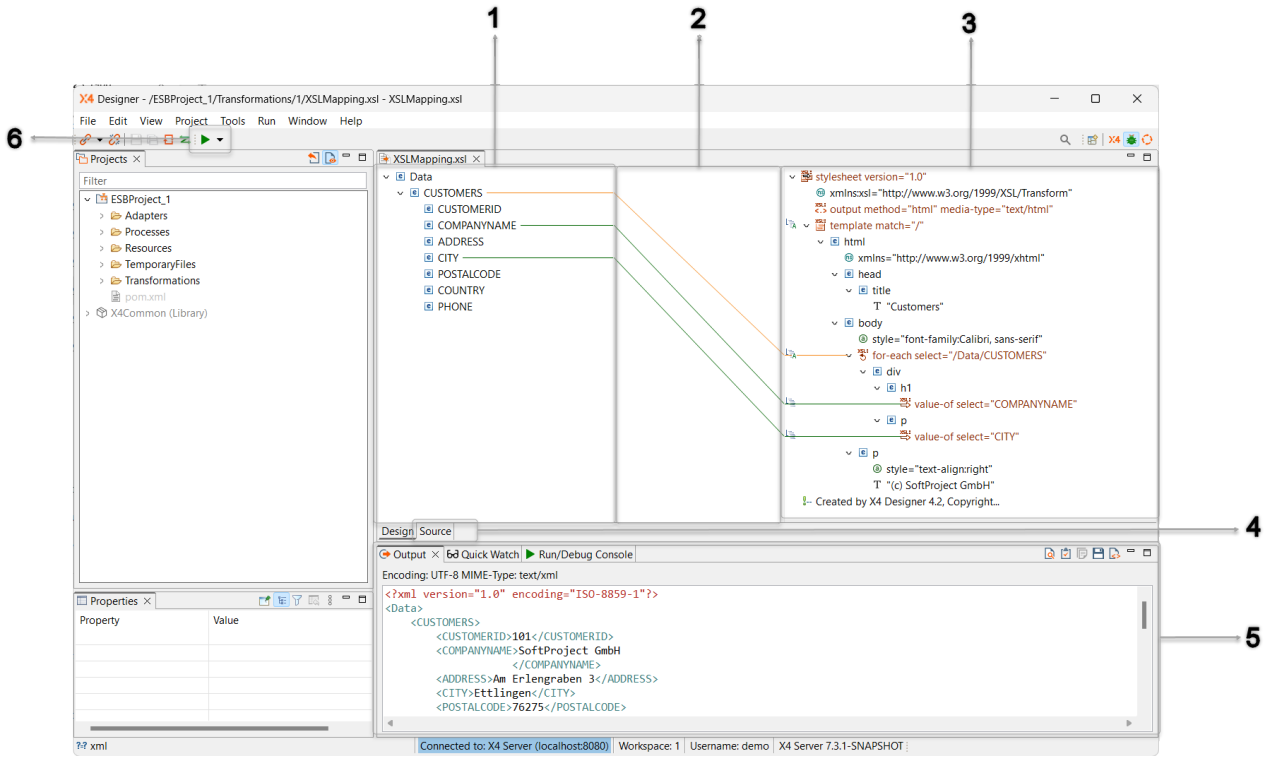
An XSL mapping involves three documents:

- XML source document (or an XML schema to provide the input structure)
- XSLT stylesheet (created with the *Mapping Editor*)
- Result document (in most cases: XML)

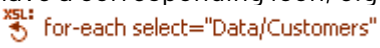
To open the Mapping Editor, create a new mapping or open an already existing XSL mapping.

The Mapping Editor's structure

The Mapping Editor is divided into the following areas:



1	**Source pane**	The source document to be transformed or an XML Schema file (`.xsd`) is displayed here. Using the context menu, you can edit selected nodes as follows: • **Copy XPath for this element to Clipboard**: copy XPath expression of the selected element to the clipboard and use it elsewhere • **Assign Input**: assign input XML document • **Remove Input**: remove assigned input XML document • **Paste Input**: insert XML document from the clipboard
2	**Linking pane**	The Linking pane displays lines that visualize the relationships between source and stylesheet nodes. ⓘ The chapter Tastenkombinationen im Mapping Editor gives you an overview of the most important key combinations helping you to work with XSL stylesheets.

3	**Stylesheet pane**	The Stylesheet pane is where you edit the stylesheet. It displays the XSL stylesheet in an editable tree with specialized features for editing XSLT stylesheets.
		
		The different parts of the stylesheet are displayed with symbols and colored highlighting in the tree view:
		
		• Nodes that contain XSLT instructions are colored brown and have a corresponding icon, e.g., 
		• If a node contains an XPath expression, it is indicated by the XPath icon on the left side. You can click on this icon to edit the XPath expression.
		• Virtual nodes (incurred nodes from an assigned XML document or XML schema that do not yet explicitly exist in the XSLT stylesheet) are greyed out.
		
		Using the context menu in the stylesheet pane, you can edit selected nodes as follows:
		
		• **Edit XSL Element**: Edit selected XSL element
		• **Rename**: Rename the node
		• **Element**: Insert an element
		• **Attribute**: Insert attribute
		• **Edit**: (In case of processing instructions and namespace declarations) Edit node content
		• **XSL Element**: Insert XSL element
		  • `as Child`: as child node
		  • `Before:` as node before the selected node
		  • `After:` as node after the selected node
		• **Add other nodes**: Add other nodes (`Namespace`, `Text`, `Comment`, `Processing Instruction`)
		• **Add here**: (In case of virtual nodes) Convert a virtual node into a real node
		• **Add Processing Instructions**: Insert processing instructions to set Dynamic Parameters.
		  • `Condition State` for status information
		  • `Log Info` for log messages
		  • `Dynamic Parameter` to set dynamic values
		  • `No Result` for empty mapping outputs
		• **Add Parameter**: Insert system, global or project parameters, see Creating and Using Parameters
		• **Add Parent**: Insert a parent element (superordinate element)
		• **Cut/Copy/Paste**: Cut, copy, or paste selection
		• **Delete:** Delete the selected node
		• **Expand/Collapse**: Expand or collapse the child nodes of the selected node
		• **Assign Virtual Nodes**: Insert virtual nodes from an XML document or schema
		• **Remove Virtual Node**: Delete virtual nodes from an XML document or schema

		The **XSL Element Creation Wizard** allows to create XSL mappings quickly and easily. The wizard offers the complete range of XSLT 2.0 elements and enables to create them easily via the context menu. Within the wizard, you can specify whether the new element is to be created before, after or as a child of the selected element. Features like auto-completion, or the display of all available and allowed XSL elements enable faster work. Moreover, the wizard allows to set and to edit the attributes of the XSL element. Required attributes are set automatically, optional attributes can be selected and filled as needed. For attributes that have multiple options to choose from, a drop-down list with all available values is provided.
4	**Source and Design view**	You can switch between two different views within the Mapping Editor: • **Design** : The editor displays the XSLT stylesheet as a tree in a three column view. Here, you can graphically map nodes of the source document/schema to the XSLT stylesheet nodes by drawing connection lines. • **Source** : In this source code view, you can edit the XSLT stylesheet directly in an XML editor
5	**Output**	The result of locally run XSL mappings is displayed in the Output view. The following additional actions can be performed via the toolbar: • *Save content of output view*: Save the data form the output view • *Format the content of the output view*: Format the data from the output view
6	**Run transformation**	Via **Run Transformation,** you can run the XSL mapping locally and display the transformation result in the output view (5).

2.5.4.1 Shortcuts in the Mapping Editor

Useful shortcuts for working with XSL stylesheets within the graphical view of Mapping Editor

Drag&Drop Actions

Reference a value to a target node using `xsl:value-of`	Drag a node from the source document to the target node
Reference multiple values into a target node using `xsl:value-of`	Keep the **Alt** key pressed and drag all desired nodes from the source document to the target node Result: An `xsl:value-of` element will be created whose `select` attribute contains all referenced nodes. In a next step, this list of nodes can be correlated by using an XPath function like `concat()` or logical operators. *Example:* `<xsl:value-of select="Element1Element2">`
Copy a node using `xsl:copy-of`	Keep the **Ctrl** key pressed, and drag a node from the source document to the target node
Create a for-each loop for a node using `xsl:for-each`	Keep the **Shift** key pressed, and drag a node from the source document to the target node

	Function	Shortcut
Basic functions	Edit element	`Enter`
	Run XSL transformation	`F9`
	Insert element	`Ins`
	Insert attribute	`Alt+Ins`
	Insert text	`Ctrl+Ins`
	Insert surrounding parent element	`Ctrl+Shift+Ins`
	Rename a element	`Alt+Shift+R`
	Remove selected node	`Del`
	Format/indent the code (in Source view)	`Ctrl+Shift+F`
Copying nodes	Insert selected node as preceding sibling	`Ctrl+ Alt+Up`
	Insert selected node as following sibling	`Ctrl+ Alt+Down`
	Insert selected node as preceding parent	`Ctrl+Alt+Left`
Moving nodes	Move selected node up	`Alt+Up`
	Move selected node down	`Alt+Down`
	Move to parent element (pull up)	`Alt+Shift+Left`
Navigating trough the stylesheet tree	Go to parent node	`Ctrl+Shift+Left`
	Go to next node	`Ctrl+Shift+Down`

	Go to previous node	`Ctrl+Shift+Up`
	Select connected nodes in the source pane and stylesheet pane	`Shift+Ins`
	Restore last selection	`Alt+Shift+Down`

2.5.5 Process Editor

With the Process Editor, Technical Processes can be modeled graphically. For each Technical Process and for the individual process modules, you can also define how they should be executed.

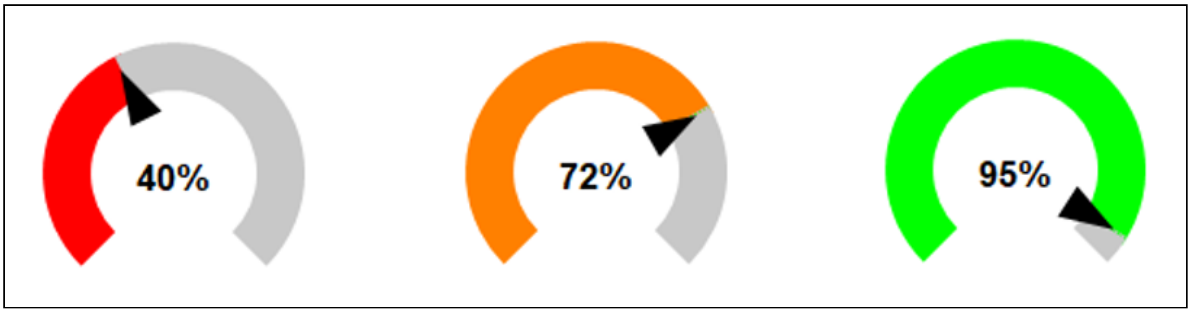
With the Process Editor you can:

- establish connections between different back-end systems,
- transform data in real time,
- implement business rules in the modeled process and
- debug modeled processes and execute them on the X4 server.

2.5.5.1 Process Quality Index

> ⓘ The Process Quality Index is an experimental feature. The functionality is currently still limited and is being developed continuously.

The Process Quality Index is a graphical tool that indicates how well an ESB process has been modeled in terms of standards compliance and clarity.



Rating of the criteria

Criteria	Condition	Rating
Number of elements	**100%** → **< 15** elements **75%** → **< 20** elements **50%** → **< 25** elements **25%** → **< 30** elements **0%** → **>= 30** elements	45%

Criteria	Condition	Rating
Number of stop elements	**100%** → **< 3** elements **75%** → **< 4** elements **50%** → **< 5** elements **25%** → **< 6** elements **0%** → **>= 6** elements	15%
Labels of the elements are not empty	**true** → **100%** **false** → **0%**	10%
Process title was set (not the name of process file)	**true** → **100%** **false** → **0%**	5%
Labels of the elements were adjusted (not filename)	**true** → **100%** **false** → **0%**	5%
No elements on the right side of stop elements	**true** → **100%** **false** → **0%**	5%
No elements on the left side of start elements	**true** → **100%** **false** → **0%**	5%
No overlapping elements	**true** → **100%** **false** → **0%**	5%
Flow Direction Index		5%
Maximum		**100%**

2.5.5.2 Shortcuts in the Process Editor

All shortcuts of the Process Editor in an overview.

	Function	Shortcut or action
Running and debugging processes	Execute process	`F9`
	Debug process (and eventually abort active debug session)	`F4`
	Execute and show a single process action (also in subprocesses) while debugging	`F5`
	Execute a process action (without showing subprocess actions) while debugging	`F6`
	Jump back from the subprocess to the main process while debugging	`F7`
	Pause debugging	`Ctrl + F8`
	Resume paused debugging	`F8`
	Cancel debugging	`Ctrl + F2`
Inserting process components and transitions	Insert process component from the repository between two connected components	Insert a file from the repository onto the transition via drag&drop; When the line expands, drop the symbol.
	Move process component while keeping the transition	Press `Alt` while moving the process component on the workplace.
	Delete process component while keeping the transition	`Alt + Del.`
	Replace process component by another repository file	Drag and drop a repository file on the process component to be replaced.
	Insert process component behind an unconnected component, and automatically create a transition	Select the unconnected process component, press `Shift`, and drag the repository file behind the selected process component.
Editing elements	Select all workplace elements	`Strg+A`
	Cut out process component	`Strg+X`
	Copy process component	`Strg+C`
	Paste process component	`Strg+V`
	Delete process component	`Del`
	Undo last action	`Ctrl+Z`
	Redo last action	`Ctrl+Y`
	Edit process component	Double-click the symbol
	Close editor	`Ctrl+W`
	Save modified element	`Ctrl+S`

	Rename repository file	F2
	Move repository file	Move the file into the target folder via drag&drop
	Reload	F5
Changing views	Zoom out of the process diagram	Ctrl+-
	Zoom into the process diagram	Ctrl++

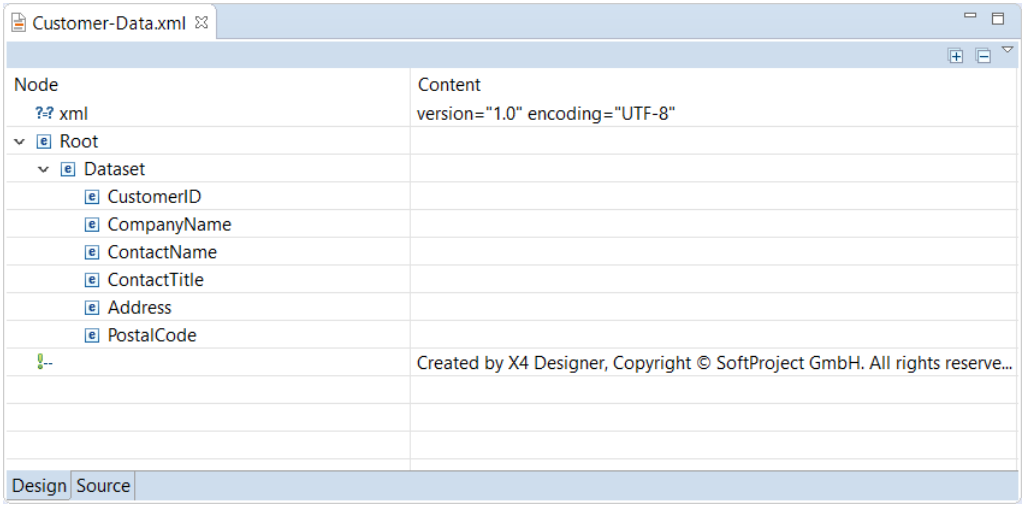
2.5.6 XML Editor

Using the XML Editor, you can create and edit well-formed XML documents.

> To open the XML Editor, create a new XML file or open an already existing file.

The Structure of the XML Editor

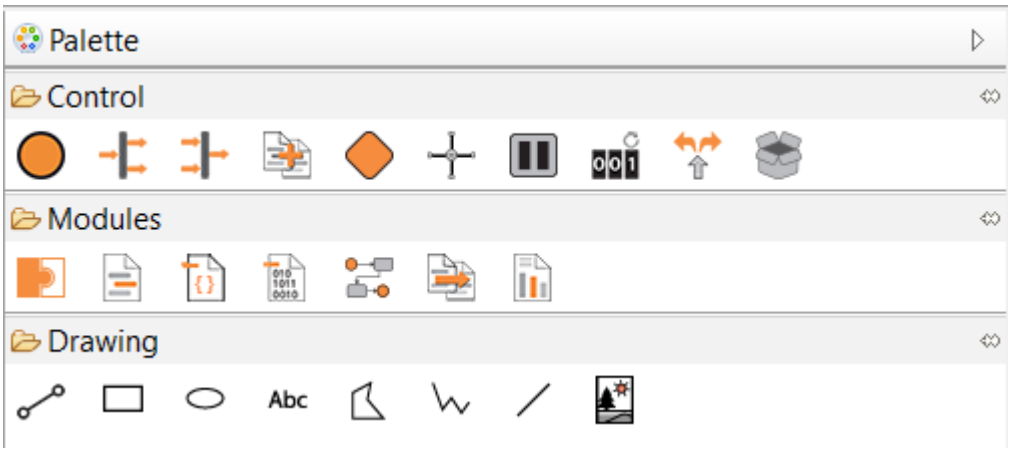
There are two views available for the processing of XML documents, the **Design** view and the **Source** view. You can switch between these to views by using the tabs at the Editor's lower edge.



Design view	The **Design** view displays the tree structure of the opened XML document as a table. In the left column, you find the different nodes, which can be expanded, if child nodes exist. In the right column, the node values are displayed. The context menu provides the following functions: • **Remove**:Remove selected node • **Add DTD Information**:Add a DTD (Document Type Definition) structure information • **Edit Namespaces**:Edit the node's namespaces • **Add Attribute**:Add new XML attribute • **Add Child**:Add child node (element, processing instruction etc.) • **Add Before**: Add node before the selected node • **Add After**:Add node behind the selected node
Source view	In the XML Editor's **Source** view, you can edit the XML document's source code. The editor supports you with the following functions: • Syntax highlighting: Different node types are highlighted with colors. • Check well-formed: XML documents are checked, if they are well-formed when saving. • Syntax completion: The editor suggests elements when you want to insert new nodes • Pretty-Print: Format the XML documents with indents and line breaks to be more readable via the **Source > Format** menu.

3 Process Components and Their Use

At the right edge of the drawing area there is a fold-out toolbar, the *palette*. It contains process module symbols that you can drag and drop into the drawing area.



If you hover over a component placed in the drawing area, various symbols are displayed above the component.

> ⚠ Not all symbols are available for all components!

Symbol	Function
+	Add new element after the current element
	Change type of the component from `Resource` to `Buffer`
	Change type of the component from `Buffer` to `Resource` ⚠ This option is not available for files located in the `TemporaryFiles` folder.
	Edit component
	Change operation of the component from `Read` to `Write`
	Change operation of the component from `Write` to `Read`
✕	Edit component

3.1 Connecting Process Components

`Transition`: Connects the individual process components and defines the sequence of process steps by its arrow direction.

Properties

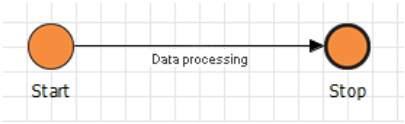
> ⚠ Some properties are only visible if the **Show Advanced Properties** option is enabled in the **Properties** view.

`Label`	Free text field for labeling the transition (for clearer design or documentation).
`Log4J Log`	Enable logging via *Log4j* at this point in the process with the current input data. To enable the logging via *Log4j*, highlight the property and click . **Possible values:** • *`Disabled:`* No Log4j action has been defined (default). • *`Enabled:`* A Log4j action has been defined.
`Save Point`	Set Save Point at this point of the process to persist the complete state of the process engine with the processed data and state information in the file system when executing a process instance. The process instance can be restarted from the last saved Save Point via the *monitoring interfaces*. **Possible values:** • *`No Savepoint`*: No Save Point has been defined (default) • *`Save and Continue`*: Create Save Point and execute the process directly • *`Save and Stop`*: Create Save Point and stop execution of the process instance • *`Delete`*: Delete current Save Point
`Source Component`	Component from which the transition line starts (is set by the X4 Designer)
`Target Component`	Component to which the transition line leads (is set by the X4 Designer)

`Priority`	For *forks* and *conditions*: position in the execution sequence (priority) **Possible values:** • any positive integer • *1*: highest priority (default)
`Condition`	For condition components: execution condition stored by the **Condition Manager** (is set by the **Condition Manager**).

Example

Transition between two process components with a label:



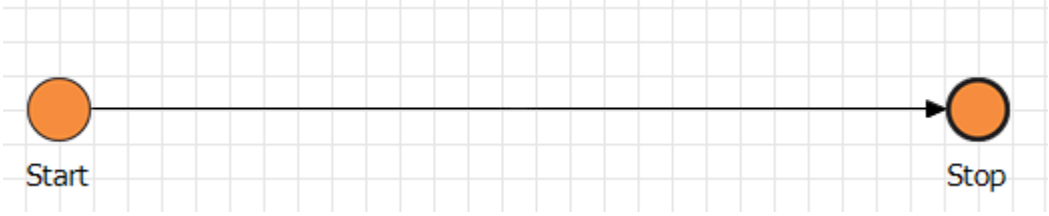
3.2 Control Components

The following control components provide the basic functions for the operation of X4 processes and their control structures:

On the following pages you will find more information about the function of the components.

3.2.1 Stop

The basic elements of each process are a defined process start (`Start`), and normally at least one process end (`Stop`).



The `Start` component cannot be removed and it is not possible to add additional `Start`-B components. Therefore, the palette only contains the `Stop` component (

) under **Controls**.

Properties and Parameters

`Status Code` (`Stop` component only)	Status code of X4 Engine. Using the `Stop` element as a ReST endpoint in a process, the HTTP status code is returned when reaching the `Stop` element. ⚠ If the **Stop on error** flag is enabled for a process, the status code 500 is returned independent of the status code of the `Stop` element. If **Stop on error** is disabled, the status code of the `Stop` element is returned. **Possible values:** • empty (default): Default status code is returned (200 = process successful/ 500 = process with error) • any numeric value (valid HTTPS status code, e.g., 200 = success/500 = with error)
`Label`	Label of the symbol within the process diagram

Input and Output

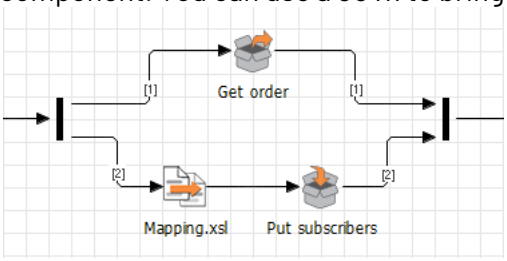
Both the `Start` and `Stop` components output the input that might come from the preceding process action (for `Start` the process input) without modification to the next process action. The `Stop` component also hands over the status of the preceding process action without modification, and uses this status as execution status for the process.

3.2.2 Fork and Join

`Fork` and

`Join`:

To execute two or more independent execution branches using the same input, you can use a `Fork` component. You can use a `Join` to bring the separate execution branches together again.



Properties and Parameters

`Rotation`	Rotation of the component in degrees **Possible values:** • *`0`*: no rotation (default) • *`90`*: rotate by 90° • *`180`*: rotate by 180° • *`270`*: rotate by 270°

Input and Output

The `Fork` component passes the input document without modification to all execution branches.

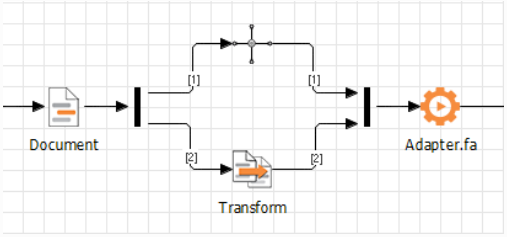
The `Join` component keeps the output documents of the last executed step of each branch in memory. If the output documents are XML documents, all output documents can be merged together in one XML document using the `Merge` component. If the branches output a binary document or an XSL mapping outputs dynamic parameters before the `Join` component, a transfer adapter can process both dynamic parameters and the binary stream.

Using Fork and Join

An execution branch can consist of one or multiple process steps or components.

The branches are not executed simultaneously, but according to the given priority that is displayed on the transitions by a number. You can set the priority of each execution branch in its properties.

Use a `Connection` component ( ) within `Fork/Join` constructs where one execution branch has an operation (e.g., an XSL mapping to dynamically parameterize an adapter) and the other branch only passes the input document.



ATTENTION: Unexpected Process Engine behavior!
If you place a loop inside another loop containing a `Fork/Join` construct, the Process Engine cannot reset the state of already executed elements correctly.
Pack the inner loop with all its process components as subprocess, see Subprocess.

3.2.3 Merge

`Merge`: Merges XML documents from different sources into one XML document, or merges text or binary streams.

Properties

`Label`	Label of the symbol within the process diagram
`Root Element`	Name of the root element that contains the entire content of the merged XML documents **Possible values:** • String that is valid for XML element names. Default: *`Merge`*

Merging Data

The `Merge` component can be placed behind the `Join` component of a fork. If XML documents are merged, the XML document generated by the `Merge` component has a root element containing the root elements and all child nodes of the merged XML documents. This XML structure can be processed by the next process action. The name of the root element can be defined using the `Root Element` property of the `Merge` component.

If two XML documents are merged, which have already been merged in a previous process step using the `Merge` component, the result XML document contains only one root element <`Merge`> and all child nodes of the merged XML documents.

Note:

- If XML documents and at least one non-XML document are merged using a Merge component, the output of all documents is concatenated in the output of all documents generated by the `Merge` component.

Merging two documents

```
<?xml version="1.0" encoding="UTF-8" ?>
<Merge>
   <!-- Document1 -->
   <!-- Document2 -->
</Merge>
```

If the input documents are text files (MIME type `text...` or `application/xml`) with the same character encoding, its encoding is used. If the encodings differ, a UTF-8 sequence is output. If all input documents have the same MIME type, the result also has this MIME type, else `text/plain`.

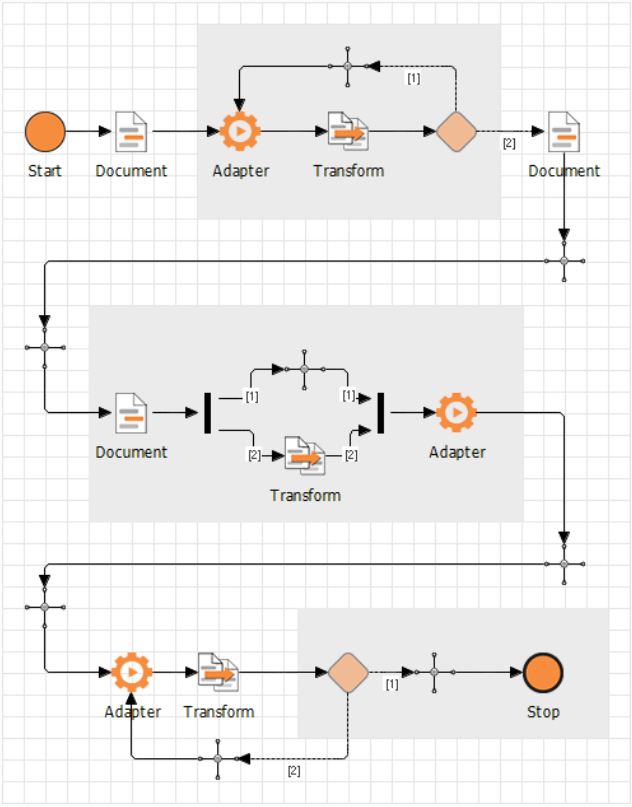
3.2.4 Condition

`Condition`: Makes case distinctions based on branches within the process, whereby each underlying branch is executed if the condition is true.

Properties

`Label`	Free text area to set a label for the condition
`Condition`	Execution condition (set automatically). Double-click the connection line (transition) to set conditions.
`Rotation`	Rotation of the `Condition` component on the workplace in degrees to display the process flow correctly depending on the horizontal or vertical process layout (also from left to right or from bottom to top) **Possible values:** • *0*: no rotation (default) • *90*: rotated by 90° • *180*: rotated by 180° • *270*: rotated by 270°

In loops, place a `Connection` component between the `Condition` component and the return to the first loop component. Also, place a `Connection` between a `Condition` component and an immediately following `Stop` component.



3.2.4.1 Creating and Using Conditions

The `Condition` component enables you to model branches in processes, where each execution branch must contain at least one condition.
To model case distinctions, you can define conditions based on:

- the execution status of previously executed process components
- parameters (global, system or project parameters)
- context variables

All set conditions are evaluated during process runtime by the X4 ESB Server's Process Engine.

If you did not set a condition for an execution branch, it will be *always* executed. This will be indicated by a semi-transparent `Condition` component. If multiple conditions apply, all matching branches will be executed according to their priority.

Prerequisites

To use the `Condition` component in Technical Process properly, the following prerequisites must be met:

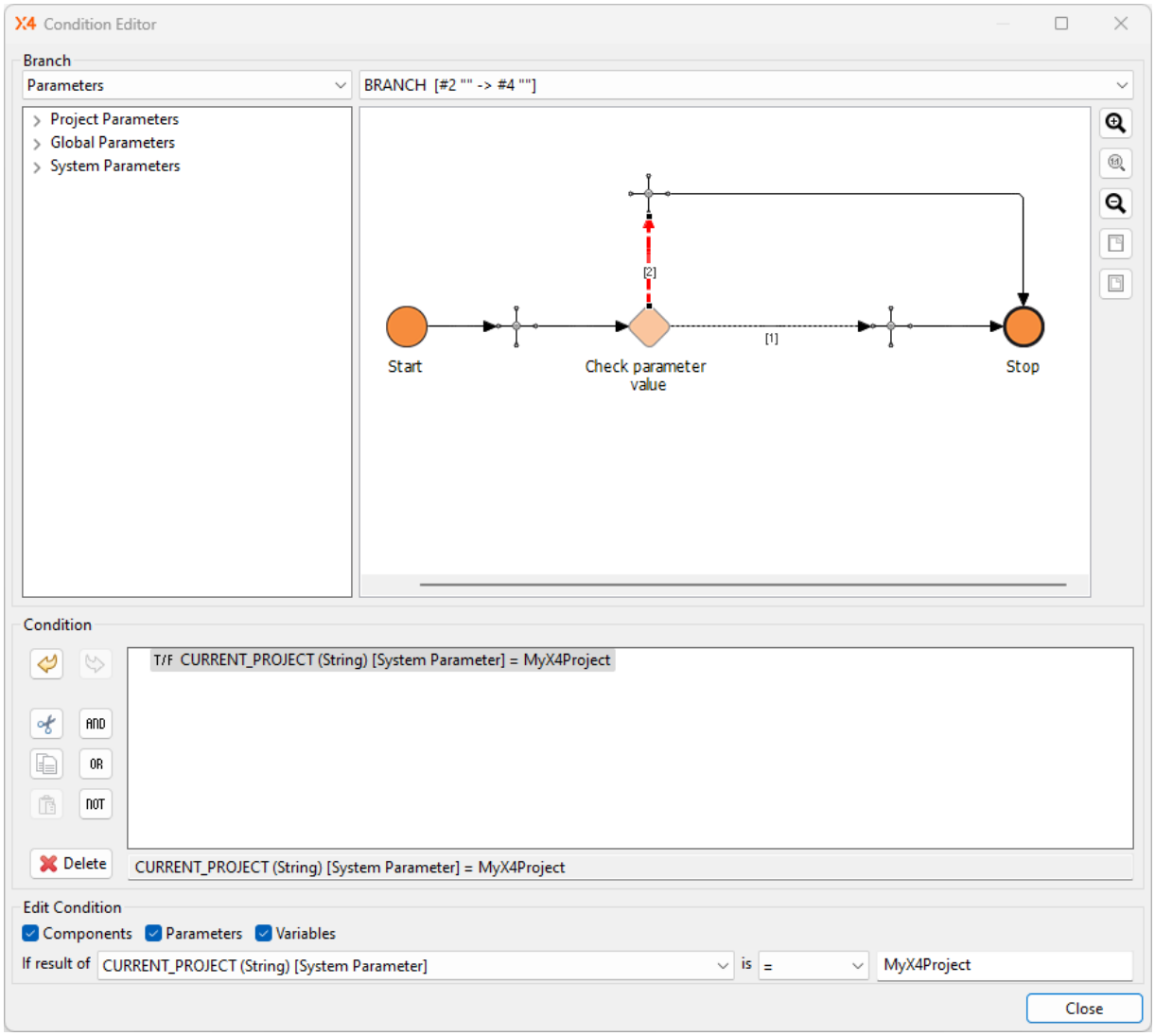
- The `Condition` component must have at least two execution branches.
- Execution branches may not be totally empty. At least one execution branch must contain a process component.
- For each execution branch, at least one condition must be defined that is explicit and valid.
- The `Condition` component should not be placed inside a fork/join construct.

Inserting a Condition Component

To insert a `Condition` component into a Technical Process, perform the following steps:

1. From the **palette**, insert a `Condition` component into the process via drag-and-drop.
2. Create a transition from the preceding process component to the `Condition` component, and repeat this for each execution branch.

3. Double-click the `Condition` component or an execution branch to open the **Condition Editor**.



4. In the **Condition Editor**, define the conditions for each execution branch.
   Thus, click on a transition behind the `Condition` component to select an execution branch. The selected transition will be highlighted red. In the **Condition** area, you can now define evaluation criteria (based on the execution status, based on system placeholders or custom placeholders, or based on context variables, see below).

   > Each condition can contain multiple condition elements that can be combined/ cascaded using logical operators (`AND`, `OR`, and `NOT`). Additional condition elements can be inserted using the corresponding buttons.

5. Click **Close** to save the conditions.

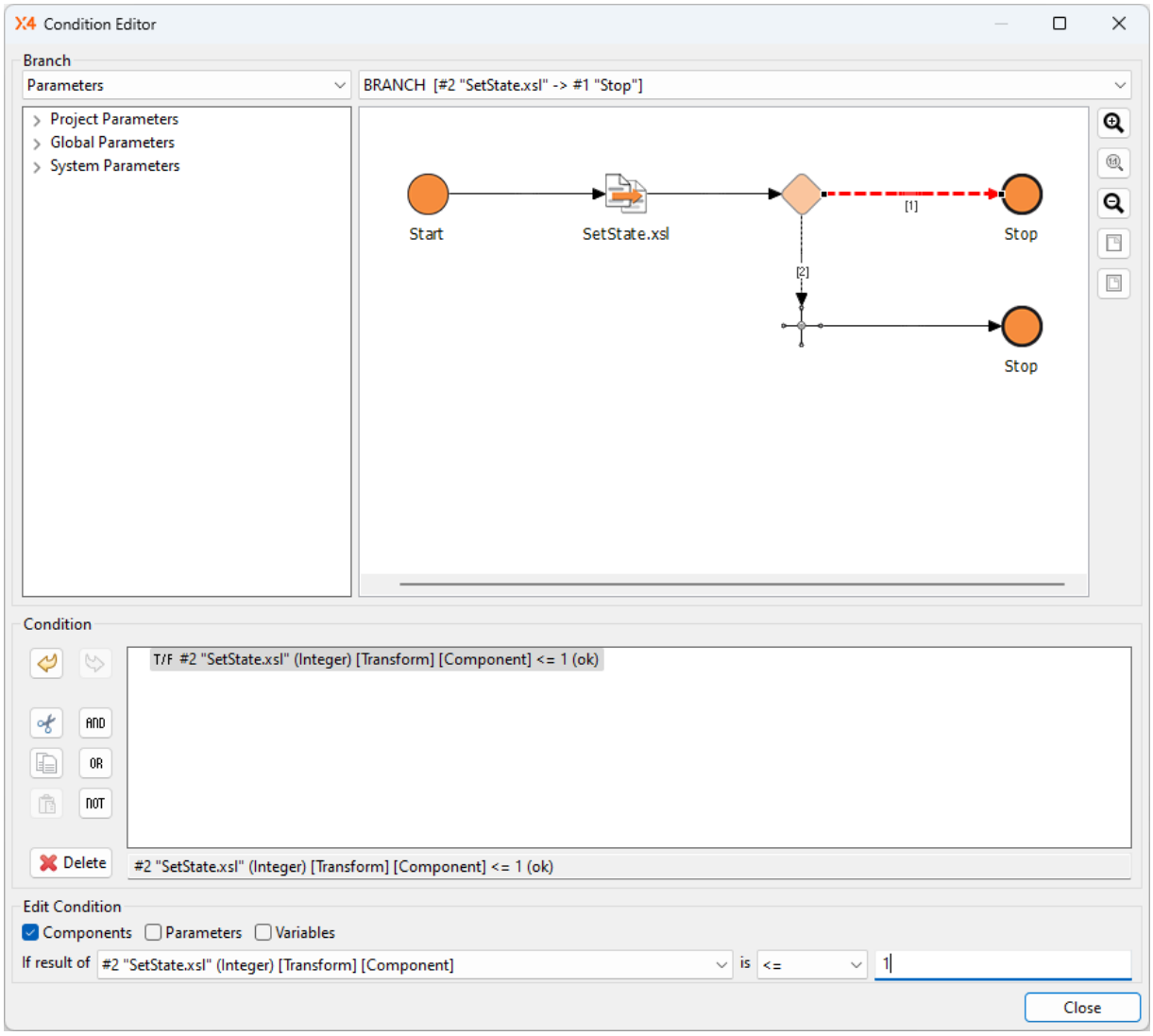
ATTENTION: Unexpected Process Engine behavior!
If you define a loop within another loop that contains a `Fork/Join` construct, the X4 Server cannot reset the status of the already executed process actions correctly.
To avoid this, put the inner loop with all its process components into a subprocess instead.
For more information, see Subprocess.

Defining Conditions Based on the Execution Status

`Condition` components enable you to define one or multiple conditions based on the *execution status* of previously executed process components. As an example, you could evaluate if an Iterator adapter does no longer output data which means that the process execution is redirected to a `Stop` component and will, thus, be ended.

To define a condition based on the execution status, perform the following steps:

1. In the **Condition Editor**, click a transition line after the `Condition` component to select an execution branch.
2. In **Edit Condition**, select the process component whose execution status should be evaluated during process runtime.
   Only preceding process components can be selected. As an alternative, you can click the corresponding component in the process diagram preview.
3. Select a comparison operator (e.g. **<**, **=**), and specify any integer number as comparison value to be evaluated, e.g. *`1 (ok)`*.



If required, you can combine/cascade conditions using logical operators (`AND`, `OR`, and `NOT`).

4. Click **Close** to save the condition.

Defining Conditions Based on System Parameters

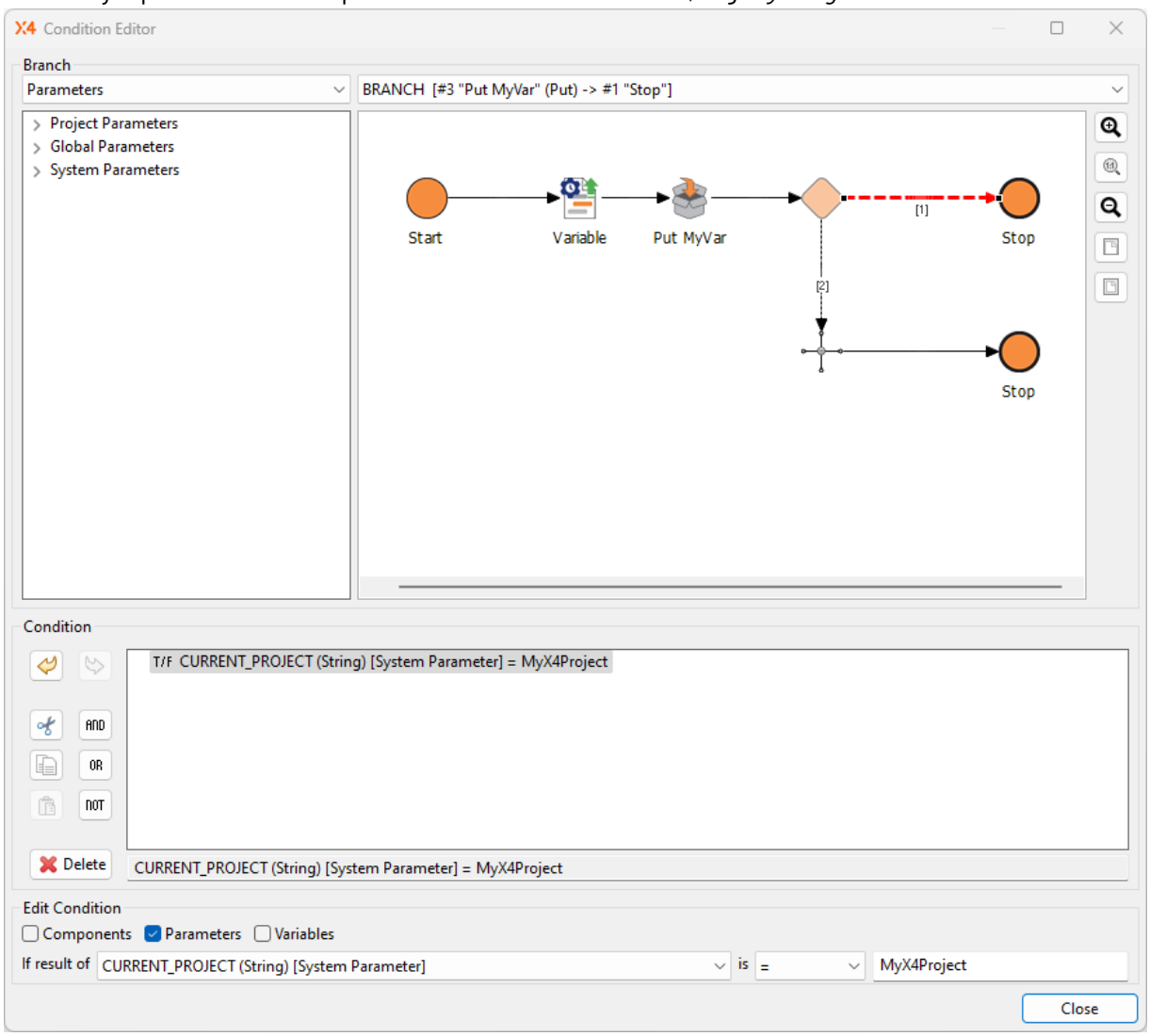
`Condition` components enable you to alter process execution based on actual environment parameters. By evaluating system parameters such as `%CURRENT_PROJECT%`, you can retrieve actual values during process runtime, see Overview of System Parameters.

To define a condition based on a parameter, perform the following steps:

1. In the **Condition Editor**, click a transition line after the `Condition` component to select an execution branch.
2. In the drop-down list on the left-hand side, choose **Parameters** to open a list below containing all available parameters and parameter groups.
3. Select a parameter to be evaluated during process runtime, e.g. *CURRENT_PROJECT.*
4. Select one of the following comparison operators:
    - = equal to ...

- `!=` not equal to ...
- `contains` contains ...
- `startsWith` starts with ...
- `endsWith` ends width ...

5. Enter any alphanumeric comparison value to be evaluated, e.g. *`MyProject`*.



If required, you can combine/cascade conditions using logical operators (`AND`, `OR`, and `NOT`).

6. Click **Close** to save the condition.

Defining Conditions Based on Global or Project Parameters

With the `Condition` components, you can use global or project parameters for the definition of conditions under certain circumstances. Just like system parameters, global or project parameters allow you to alter process execution based on actual environment parameters.

In order to use global or project parameters in conditions in the **Condition Editor** and to be able to evaluate them at process runtime, you basically proceed as when defining conditions with system parameters.

Defining Conditions Using Global Parameters

Please note the following additional prerequisites for global parameters:

- The global parameter must have already been defined within the **Global Parameter Configuration** in order to be available in the **Condition Editor.** For more information, see Global Parameters and Groups.
- The global parameter must also be available in the **Global Parameter Configuration** during process runtime, in order to be evaluated by X4 ESB Server's Process Engine.
- In process property `Placeholder Groups`, the corresponding parameter group must have been selected. For more information, see Global Parameters and Groups.

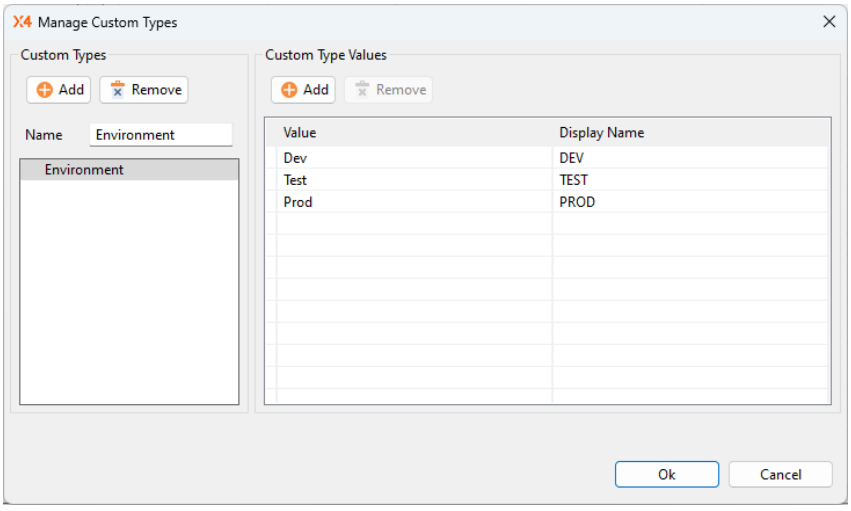
Defining Conditions Using Project Parameters

Please note the following additional prerequisites for project parameters:

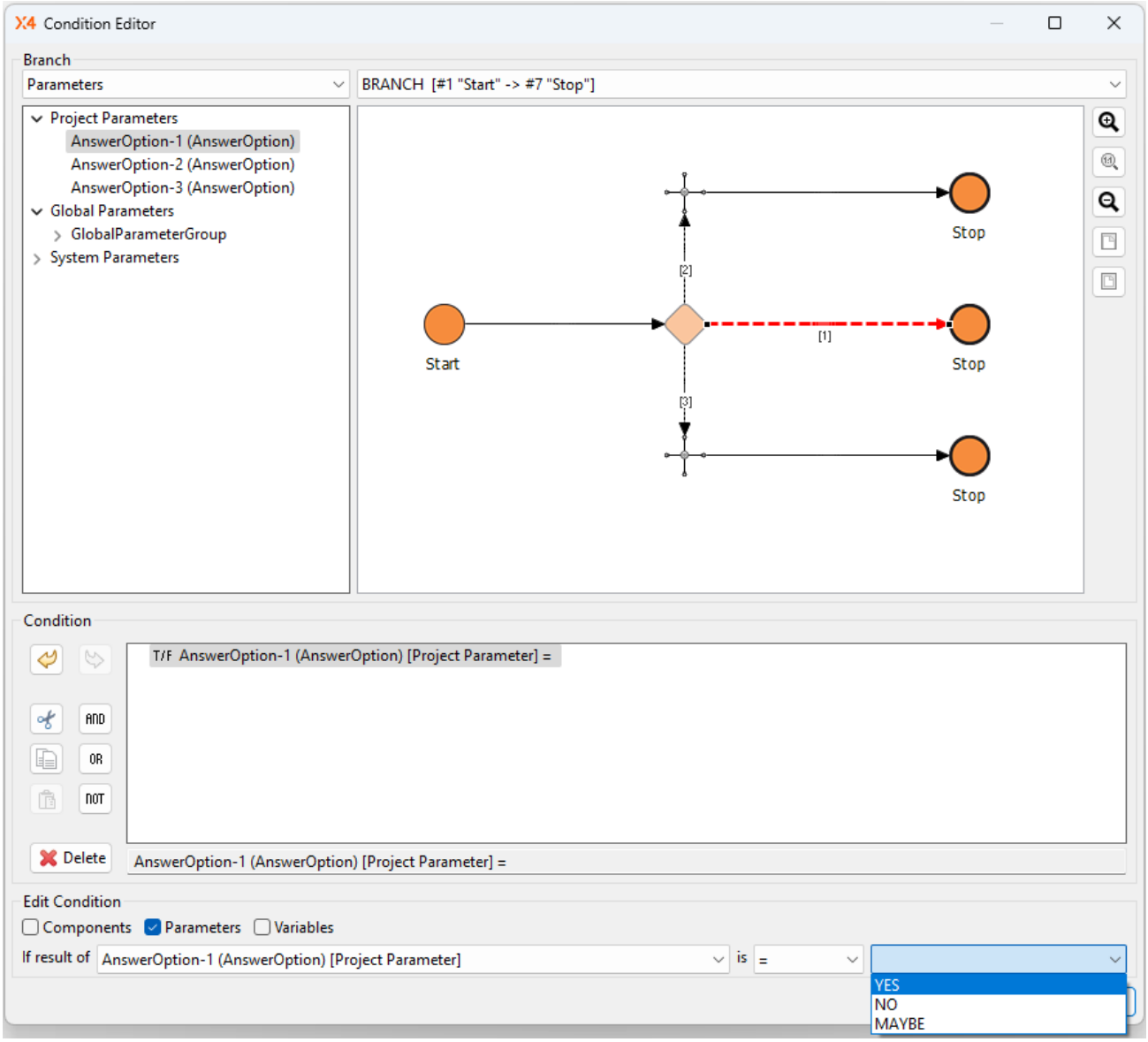
- The project parameter must already be defined in the file with the **Project Parameter Configuration** `*.param` in the project in order to be selected in the **Condition Editor**. For more information, see Set project-specific parameters.
- **Custom Types** and **Custom Type Values** can be used to define selection options for use in the **Condition Editor**. For more information, see Create custom types.

Use Case for Project Parameters with Custom Types

The **Custom Type** `Environment` has the **Custom Type Values** `Dev`, `Test`, `Prod` and the **Display Names** `Dev`, `Test`, `Prod`:



The **Values** can then be selected in the **Condition Editor**:



Defining Conditions Based on Context Variables

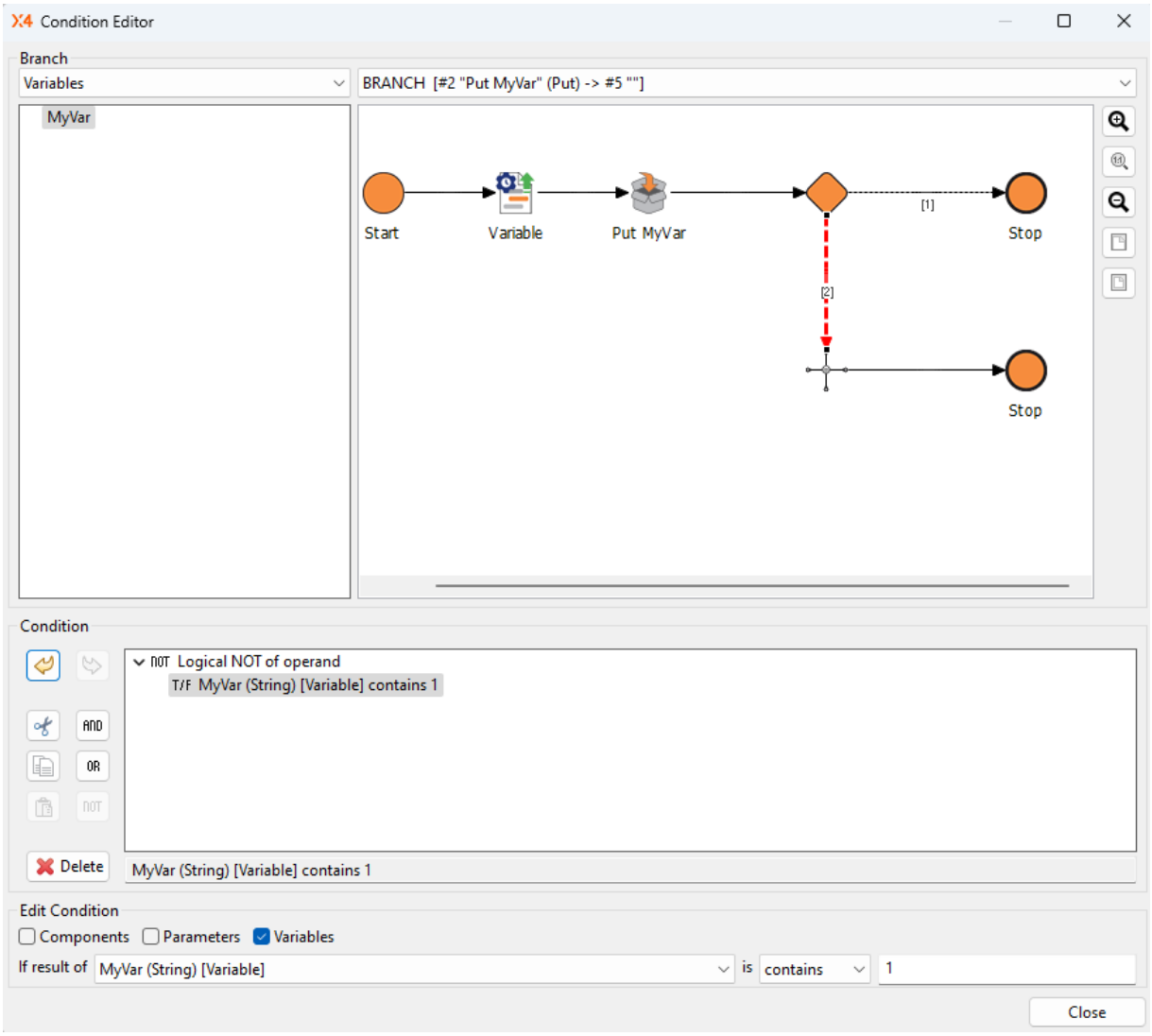
`Condition` components enable you to alter process execution based on actual context variable values.

To define a condition based on a context variable, perform the following steps:

1. In the **Condition Editor**, click a transition line after the `Condition` component to select an execution branch.
2. In the drop-down list on the left-hand side, choose **Variables**, to open a list below containing all context variables that were already defined in the process diagram.
3. Select a context variable to be evaluated during process runtime, e.g. *`MyVar`*.
4. Select one of the following comparison operators:
    - = equal to ...
    - != not equal to ...

- `contains` contains ...
- `startsWith` starts with ...
- `endsWith` ends width ...

5. Enter any alphanumeric comparison value to be evaluated, e.g. *1*.



If required, you can combine/cascade conditions using logical operators (`AND`, `OR`, and `NOT`).

6. Click **Close** to save the condition.

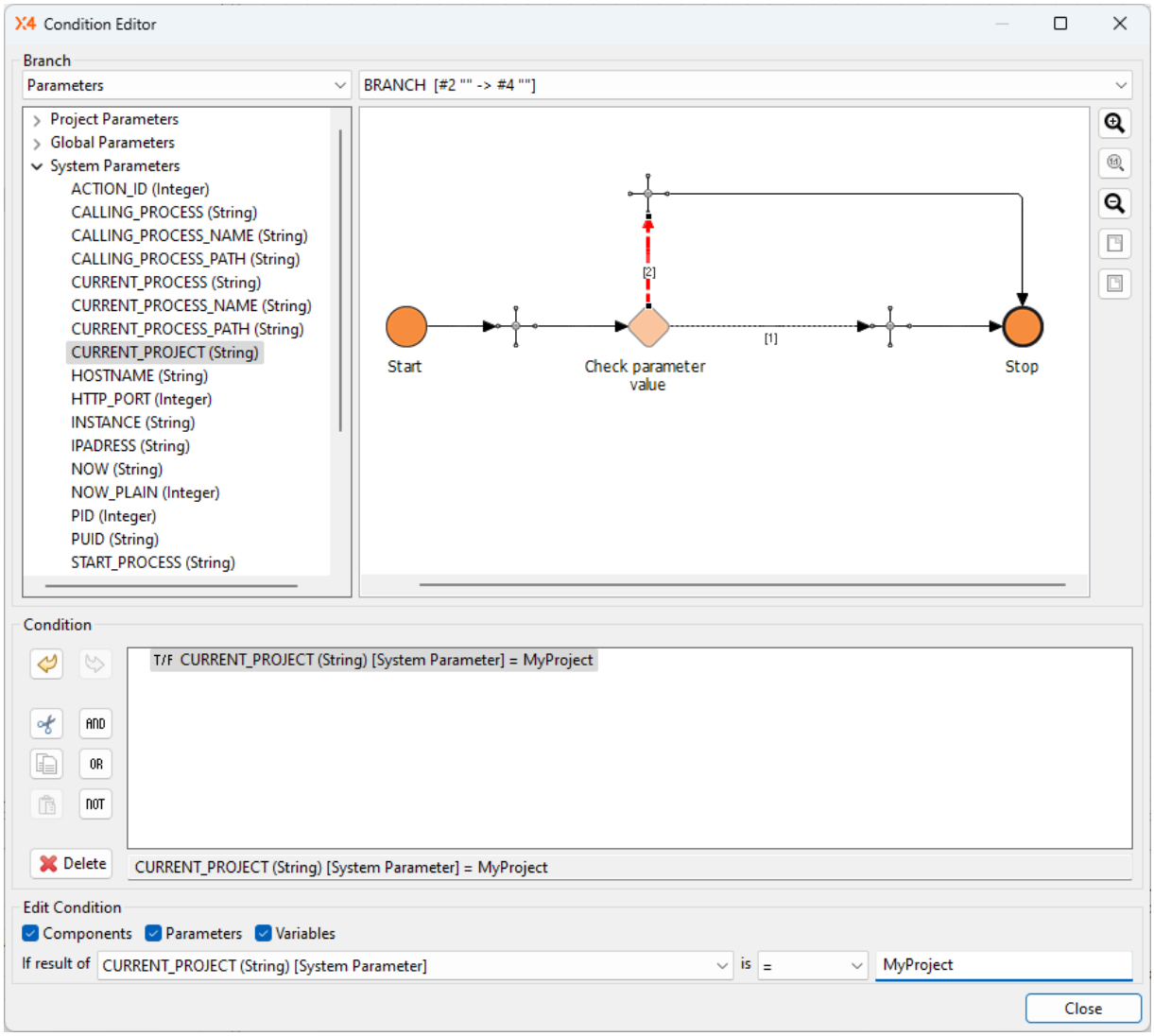
3.2.4.2 Condition Editor

The **Condition Editor** enables you to edit `Condition` components within a Technical Process, when different branches will be executed. For each branch, you can define conditions that evaluate status codes of preceding process components, context variable values, or parameter values (global, project or system parameters).

How to open the Condition Editor

1. Open a Technical Process.
2. Double-click a *Condition* symbol or one of its execution branches.
   The **Condition Editor** displays the process diagram in a simplified way without drawing

elements:



Right-hand side of the Branch area

In the **Branch** area on the right-hand side, you can select the branch for which you want to create a condition from the drop-down list. Alternatively, you can click on the correspondent transition line in the process preview.

View options for process preview:

	Zoom in	Zoom into the process diagram
	Original zoom	Display the process diagram in original size
	Zoom out	Zoom out of the process diagram

	Zoom to page	Display all
	Zoom to width	Display in full width

Left-hand side of the Branch area

In the **Branch** area on the left-hand side, you can select additional condition criteria from the drop-down list. The area below displays all entries that correspond to your selection. Choose between the following options for a condition criterion:

- **Parameters:** a list of currently available parameters and parameter groups (system, global and project parameters)

  > Global parameters can only be selected if they have previously been created in the Global Parameter Configuration.
  > Project parameters can only be selected if they have previously been created in the `*.param` file (**Project Parameter Configuration**), see also Creating project parameters.

- **Variables:** a list of context variables that have already been defined within the current Technical Process

Condition area

The **Condition** area is used to display and edit the conditions for the selected branch.
To create or edit a condition, double-click into this area.

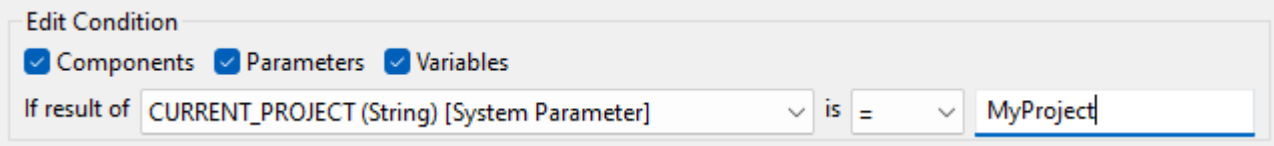
Editing options for conditions:

	Undo	Undo the last editor action
	Redo	Redo a previously undone editor action
	Cut	Cut a selection
	Copy	Copy a selection
	Paste	Paste a selection
AND	**Logical AND**	Insert logical `AND` connection

OR	**Logical OR**	Insert logical `OR` connection
NOT	**Logical NOT**	Insert logical `NOT` connection
Delete	**Delete**	Delete a selected condition

Edit Condition area

In the **Edit Condition** area, you can create or edit a selected condition. For more information, see Creating and Using Conditions.



Components	Option to specify whether **If result of** drop-down list should display process components to evaluate their execution state during process runtime
Parameters	Option to specify whether **If result of** drop-down list should display parameters (system, global or project parameters) to evaluate their actual values during process runtime
Variables	Option to specify whether **If result of** drop-down list should display context variables to evaluate their actual values during process runtime
If result of	Here you can select the criterion (process component, parameter, or variable) for which you want to define a condition.
is	Here, you can select a relational operator: • < less than... • <= less than or equal to... • = equal to... • >= greater than or equal to... • > greater than... • <> not equal to... The following comparison operators for character strings are available here for parameters and variables: • = equal to... • `!` = not equal to... • `contains` contains... • `startsWith` starts with... • `endsWith` ends width...

(Comparison value)	Enter a value which should be compared to the selected criterion. a) For processes elements, always enter a numeric status value: • `1`: Default value if executed successfully • `-1`: Default value if an error occurred • `0`: Default value if the result is empty or the process element has not yet been executed successfully • nnnn: Any integer value (e.g., 2000) related to a ConditionState Mapping b) For context variables and parameters, enter an alphanumeric value: • Content of a context variable previously generated with `Put` • Content of parameter (system or previously defined global or project parameter)

3.2.5 Connection

`Connection`: Passes the input and the execution state of the preceding process action without modification. It marks the execution branch explicitly for the X4 Server. Within a loop, this component marks the return to the first loop component. It connects a `Condition` component with a `Stop` component, or a `Fork` component with an immediately following `Join` component.**Properties**

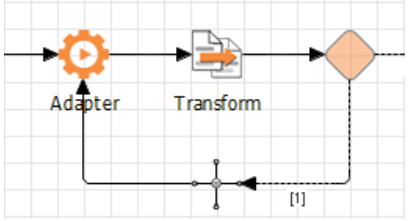
`Label`	Text label of the symbol in the process diagram

Output

The `Connection` component outputs the input without modification, including the execution status of the preceding process action.

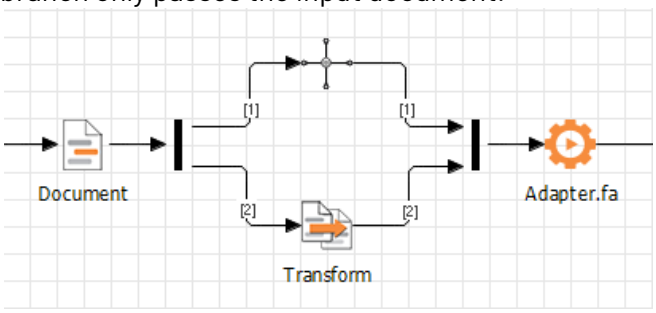
Examples

- Within loops, a `Connection` component is used as the last loop component.

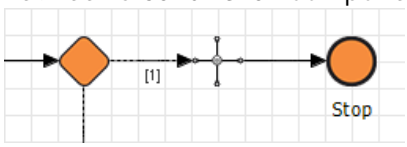


- A `Connection` component is used within `Fork/Join` constructs where one execution branch has an operation (e.g., an XSL mapping to dynamically parameterize an adapter) and the other

branch only passes the input document.



- Between a `Condition` component and a `Stop` component, a `Connection` component is used.



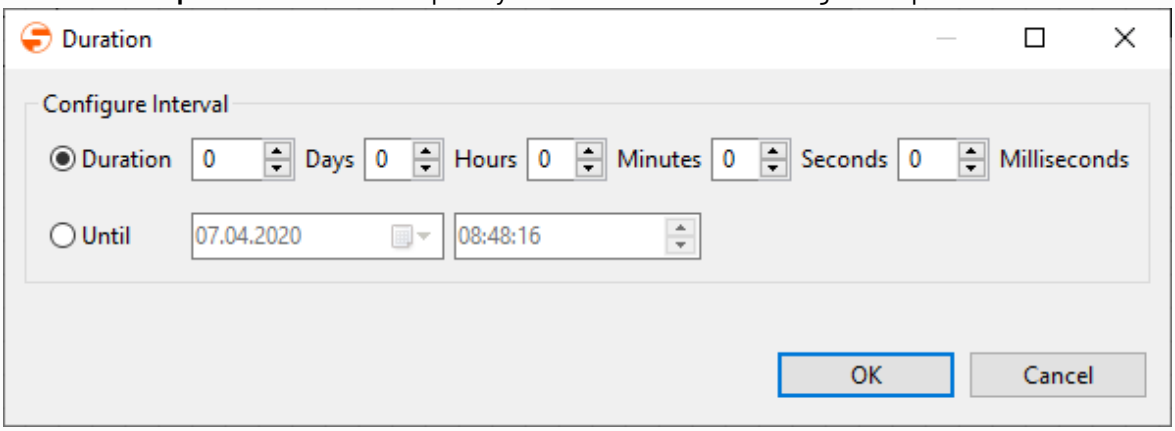
3.2.6 Pause

`Pause`: Defines a waiting time in a process; After a specified duration or on a specific date and time, the process execution is resumed. The `Pause` component receives an input by the preceding process action, and outputs it without any changes to the following process action.

Properties

`Label`	Label of the symbol within the process diagram
`Duration`	Pause duration **Possible values:** • Any duration specifying days, hours, minutes, seconds, and milliseconds • *`9999d 23h 59m 59s 999ms`*: maximum waiting time of the `Pause` component • *`0d 0h 0m 0s 0ms`*: no waiting time (default) Alternatively, you can define an end date for the pause in the format `yyyy-MM-dd hh:mm:ss` using the option `Until`, e.g., *`2010-08-30T01:55:10`*)

To define a pause duration, double-click the `Pause` component or click on `Duration` > within the **Properties** view. Then specify the duration in the dialog that opens.



3.2.6.1 Dynamically Parameterizing the Pause Component

The `Pause` component can also be parameterized dynamically using its technical parameters `sleep_time` and `sleep_until`.

- **Dynamic parameterization via `sleep_time`**
  In the following example, the `Pause` component is parameterized to a pause duration of 5 seconds:

Dynamical parameterization via 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>
```

- **Dynamic parameterization via `sleep_until`**
  In the following example, the `Pause` component is parameterized to the current date and time plus 1 minute and 15 seconds:

Dynamic parameterization via 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>
```

3.2.7 Counter

`Counter`: Enables to implement counting loops in processes by setting a start value and an end value. For each counter cycle, the value is incremented or decremented by the value entered in `Step Size`, until the end value is reached. Moreover, the properties can be set dynamically using an XSLT mapping, and the current index can be stored in a variable using `indexName`.

Properties

`Start` Dynamic parameter: `start`	Initial value from which is decremented or incremented **Possible values**: any integer greater than `0` ⓘ The start value is not counted during the loop. Note that the start value (start) can only be parameterized dynamically during the first loop.
`End` Dynamic parameter: `end`	End value up to which is decremented or incremented **Possible values**: Any integer greater than `0`. If the end value is less than the start value, the process counts down. Otherwise, the process will count up. ⓘ The end value is not counted during the loop.

`Step Size` Dynamic parameter: `step`	Counting steps for incrementing or decrementing, e.g., *2* **Possible values**: any integer greater than `0`
`Label`	Label of the symbol within the process diagram
`Index Name` Dynamic parameter: `indexName`	Name of the variable **Possible values:** name of the variable in order to access the current index of the counter

Input

The component expects an input, if the properties are set dynamically.

Output

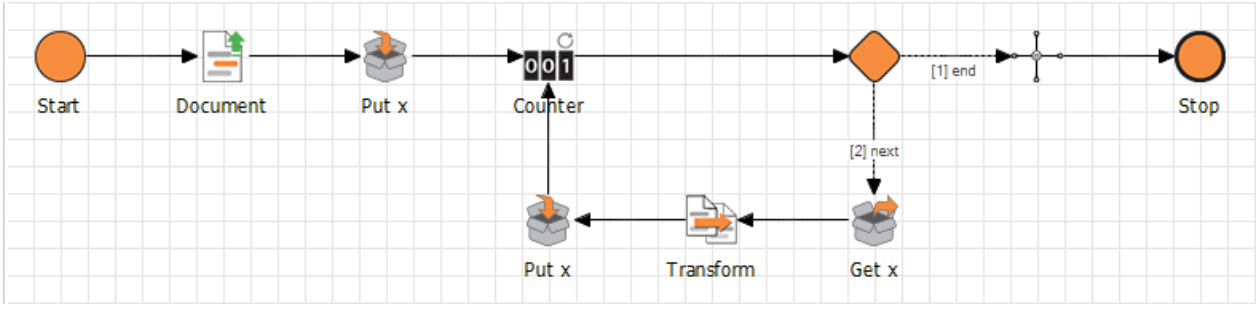
If the property is set dynamically, the component outputs a variable with *<indexName>* and the value of the current index. Otherwise, the component does not produce any output.

Status values

`0 (counting)`	Counter is running (end value was not yet reached)
`1 (finished)`	Count-down is finished (end value was reached)

Example

Simple counting loop with a `Counter` component and a `condition` component. The properties `Start=1`, `End=4`, and `Step Size=1` were set. The counting loop is repeated twice, because the start value and the end value are not counted.



3.2.8 Transaction

`Transaction`: Executes process actions as transaction: On a *Commit*, all operations are applied to the target system. On a *Rollback*, all operations are reset to the original state before the transaction. To encapsulate process actions into a transaction, place the corresponding transaction components before and after the process actions. In most cases, the transaction component is used for JMS queue operations and for database operations.

Please observe the following notes:

- Transactions do not work in the *X4 Designer*'s debug mode.
- From within a transaction, you can execute subprocesses.
- If a subprocess is started from within a transaction using the Process Starter adapter, the transaction does not apply for this subprocess (transactions do not work throughout threads).
- It is not possible to nest transactions in transactions.
- In a transaction, you can use multiple database connections and JMS queues.
- The used database connection (JDBC DataSources) must support transactional handling.
- To be able to execute transactions, the `jta` attribute must not be set to `false` when configuring the datasource to be used within the `standalone.xml`.

Properties

`Label`	Label of the symbol within the process diagram
`Operation`	Define a transaction operation **Possible values:** • *`Not assigned:`* no transaction assigned • *`Begin`*: begin transaction • *`Commit`*: execute/complete transaction • *`Rollback`*: cancel transaction and revert the transaction's operations

Status values

`1`	The transaction component was executed successfully.
`0`	The transaction component was executed without result.
`-1`	An error occurred during the transaction.

3.2.9 Variable

`Variable`: Stores data temporarily at runtime in the process's context. From within the same Technical Process or a subprocess you can access this variable. Using the `context:` protocol, you can access variables also in XSL mappings.

Properties

`Operation`	Defines if the variable is initialized, read or removed **Possible values:** • *`Put`*: Save input data from the preceding process action into the variable (default) • *`Get`*: Read the content of the variable, and output it to the next process action • *`Remove`*: Discard variable (delete it from memory) • *`Test`*: Checks whether a variable is set. If no variable is set, the status *`0`* will be output.
`Name`	Unique name of the variable **Possible values:** • Any string containing alphanumerical *US-ASCII* characters (*`Aa-Zz`* and *`0-9`*) and underscores (_), and which does not start with a number (e.g., *`Report_Data01`*) • *`X4STATE:`* Contains the status of the preceding process action as XML document. See *Context Variables*. • *`X4PREVACTION:`* Contains action ID, text label, and status of the preceding process action as XML document

3.2.9.1 Using Variables

Context variables temporarily store data in the context of a currently executed Technical Process. Each context variable must be initialized first; i.e., it requires a unique name and a value is assigned (operation `Put`). Later in the Technical Process, the context variable can be retrieved using the operation `Get`. Finally, using the `Remove` operation, you can delete variables from the memory.

Create variable component	Drag a variable component from the **Palette** and drop it into the Technical Process, then enter a unique variable name (e.g., `Var1`) within the **Properties** view in the **Name** property.
Initialize variable	In the **Properties** view in property `Operation`, choose the value `Put` to load input data from the preceding process action into the variable.
Read variable	Insert a new variable component with the identical name, and set the property `Operation` to `Get`.
Test variable	Insert a new variable component with the identical name, and set the property `Operation` to `Test`.

Delete variable from memory	Insert a new variable component with the identical name, and set the property `Operation` to `Remove`.
Access process variables from within XSL mappings	Use the `context://` URL protocol, e.g., via the XPath expression `document('context://Var1')`

3.2.9.2 Context Variables

By default, Technical Process provide a number of context variables providing information to the currently executed process instance. You can retrieve these variables either within the process with a `Variable`component using the operation `Get`, or within XSL mappings using the `context: URL protocol`.

X4STATE	Outputs the status of the previous process step (immediately before the current process step) in an XML document **Example** `<state>-1</state>`

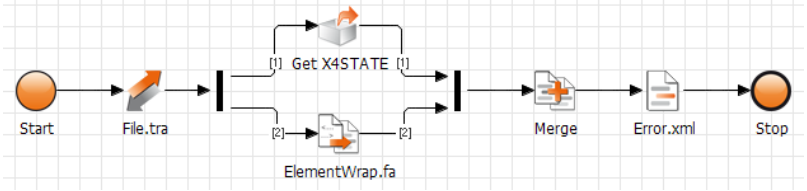
X4PREVACTION	Outputs information (action ID, text label, and status code) about the preceding process step in an XML document
	Example
	`<action>`
	`  <id>4</id>`
	`  <label>File.tra</label>`
	`  <state>-1</state>`
	`</action>`

Example

These context variables allow advanced error handling. Thus, you could output for example the error message text of an adapter together with the adapter's status.

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

Adapter error message and its status code merged in an XML document:



3.3 Module Components

The following module components connect processes with the outer world and other processes, e.g., with database connections, function adapters and subprocesses. Module components are provided as reusable files in the **Projects** view. The modules' configuration is either stored in the correspondent adapter or XSL stylesheet (for mappings and reports).

On the following pages, you will find more information on the function of the individual components.

3.3.1 Adapter

`Function Adapter`: Provides basic or additional features, such as encoding conversions, or enables you, for example, to access a remote SAP ERP system. Since X4 Designer mainly works with XML data, in most cases function adapters produce or modify XML documents. Most function adapters output XML data that can be processed in further process actions.

> To choose a function adapter type and to specify parameters, double-click the corresponding function adapter icon in the drawing area. The individual signification of the following function adapter properties is described in the respective adapter reference.

Properties

`Adapter type`	Name of the adapter
`Component Parameters`	Set predefined system, global or project parameters. For more information, see Using Component Parameters.
`Document`	Path to the linked adapter in the repository
`Label`	User-defined text visible in technical process.

`Operation`	Specifies which operation is executed by the function adapter. Values and behavior differ depending on the adapter type.

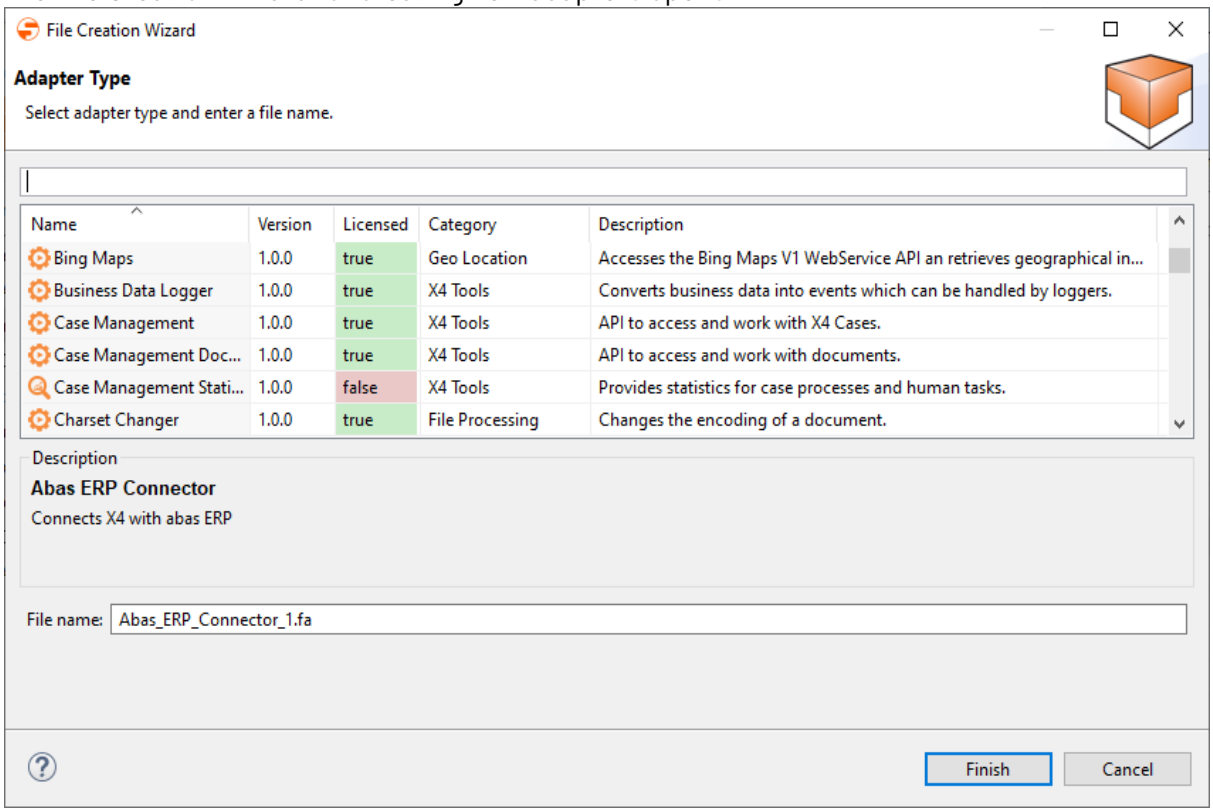
Creating and Using Adapters

Adapters can be created within an ESB project in the `Adapters` folder. There are two options available for this purpose:

- via the menu **Project > Add Adapter**
- via the context menu **New > Adapter**

To create an adapter, proceed as follows:

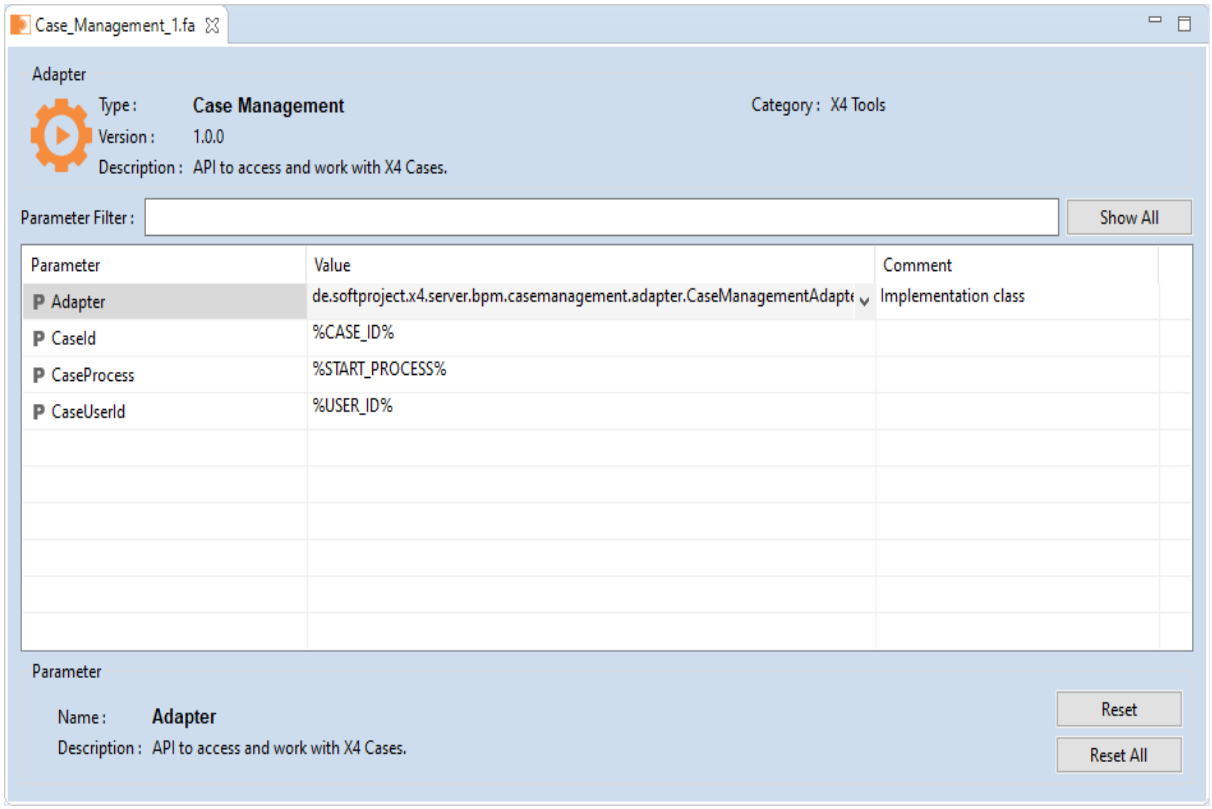
1. Right-click on the `Adapters` folder within the ESB project to open the context menu.
2. In the context menu, select **New > Adapter**.
   The File Creation Wizard for creating new adapters opens.



3. Select the desired adapter from the list.

   The search box allows you to search for adapter names and descriptions, and to filter the displayed adapters. Additionally, you can sort adapters by category by clicking on the column title `Category`.

4. Enter the adapter name in **File Name**.
5. Click **Finish** to save the adapter to the repository.
   The adapter will be opened in the Adapter Editor and can be configured.



6. Define the parameters within the Adapter Editor, at least the mandatory parameters. Mandatory fields are marked with the symbol
   P
   .

   - Set all adapter parameters that contain confidential information, dynamically with XSL mappings, e.g., using the processing instruction `DynamicParameter` (see Using Component Parameters). Thus, user credentials are not set to the adapter until the Technical Process is running. The corresponding parameters can be left blank, and confidential data is not visible in the process documentation.
   - A parameter reference for each adapter can be found in the documentation, see Overview of X4 Adapters.

7. Save the adapter.
8. Insert the adapter component from the **Projects** view into the Technical Process via drag-and-drop.
9. Select an adapter operation for the selected adapter in `Operation` within the **Properties** view. The adapter is now ready to use.
10. Save the Technical Process.

3.3.2 XML

`XML`: Includes well-formed XML data in the process. The character encoding is determined from the XML declaration and the MIME type from the file name extension.

Properties

`Document`	Path to the linked XML document in the repository
`Label`	Label of the symbol within the process diagram
`Operation`	Determines if the file is read or written **Possible values:** • *`Read`*: read XML document into the process and pass data to the next process step (default) • *`Write`*: save XML document with the data from the previous process step
`Type`	Defines the type of the file **Possible values:** • `Resource` • `Buffer` (default)

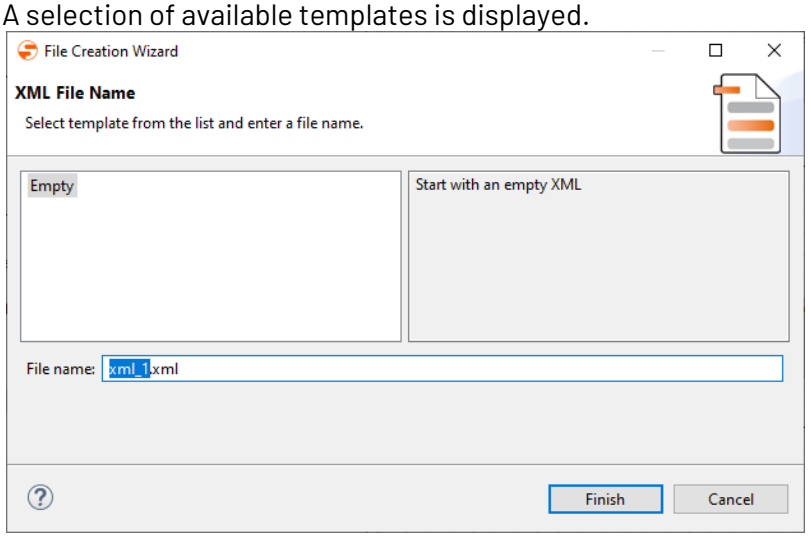
Creating and using XML document

XML documents can be created within an ESB project in the `Resources` folder (for resources) and in the `TemporaryFiles` folder (for buffer files). Two options are available for this purpose:

- via the **Project > Add XML Document** menu
- via the **New > XML Document context** menu

1. Within the ESB project, right-click on the folder to access the context menu.

2. In the context menu, click **New > XML Document**.
   A selection of available templates is displayed.



3. Select a blank template or an existing template.
4. In **File Name,** enter the file name.
5. To create the XML document, click **Finish**.
   The XML document is created and opened in the XML Editor.
6. Edit the XML document.
7. Drag and drop the XML document from the **Projects** view into the process diagram.
   Example: Insert between function adapter and `stop` module.
8. In the **Properties** view, in `Operation`, specify if the XML document should be read (*`Read`*) or written (*`Write`*).

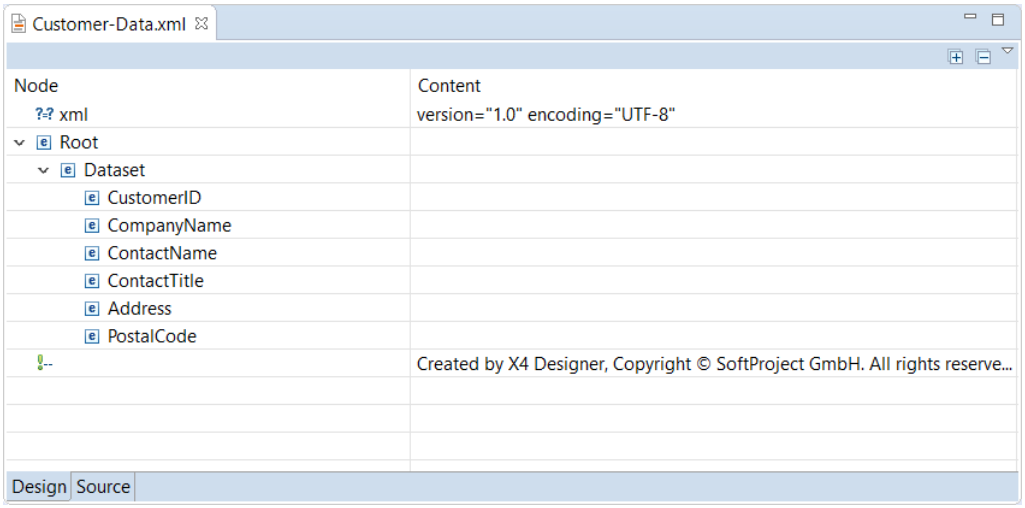
3.3.2.1 XML Editor

Using the XML Editor, you can create and edit well-formed XML documents.

> To open the XML Editor, create a new XML file or open an already existing file.

The Structure of the XML Editor

There are two views available for the processing of XML documents, the **Design** view and the **Source** view. You can switch between these to views by using the tabs at the Editor's lower edge.



Design view	The **Design** view displays the tree structure of the opened XML document as a table. In the left column, you find the different nodes, which can be expanded, if child nodes exist. In the right column, the node values are displayed. The context menu provides the following functions: • **Remove**:Remove selected node • **Add DTD Information**:Add a DTD (Document Type Definition) structure information • **Edit Namespaces**:Edit the node's namespaces • **Add Attribute**:Add new XML attribute • **Add Child**:Add child node (element, processing instruction etc.) • **Add Before**: Add node before the selected node • **Add After**:Add node behind the selected node
Source view	In the XML Editor's **Source** view, you can edit the XML document's source code. The editor supports you with the following functions: • Syntax highlighting: Different node types are highlighted with colors. • Check well-formed: XML documents are checked, if they are well-formed when saving. • Syntax completion: The editor suggests elements when you want to insert new nodes • Pretty-Print: Format the XML documents with indents and line breaks to be more readable via the **Source > Format** menu.

3.3.3 JSON Document

`JSON document`: Includes JSON documents into the process.

Properties

`Document`	Path to the linked JSON file in the repository

`Label`	Label of the symbol within the process diagram
`Operation`	Determines if the file is read or written **Possible values:** • *`Read`*: read file into the process and pass data to the next process step (default) • *`Write`*: save file with the data from the previous process step
`Type`	Defines the type of the file **Possible values:** • `Resource` • `Buffer` (default)

Create JSON Document

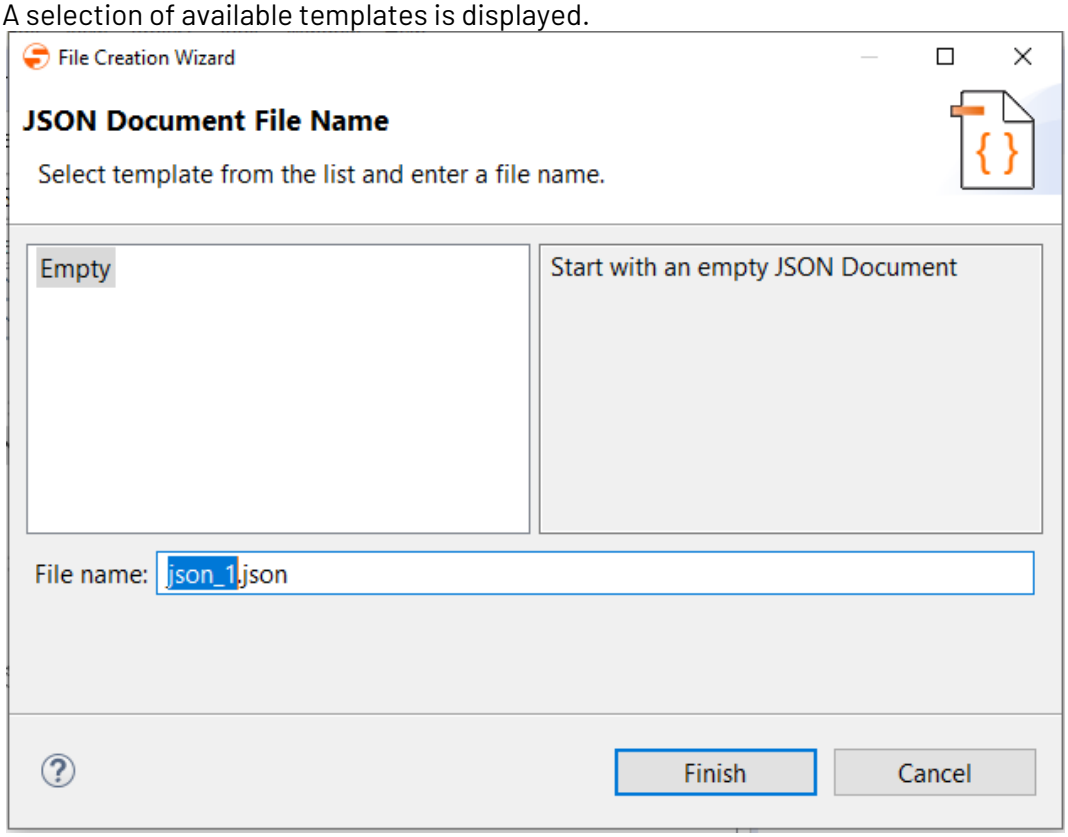
JSON documents can be created within an ESB project in the `Resources` folder (for resources) and in the `TemporaryFiles` folder (for buffer files). Different options are available for this purpose:

- via the **Project > Add JSON Document** menu
- via the **New > JSON Document** context menu
- via the **New > Add JSON Document** menu

To create a JSON document, do the following:

1. Within the ESB project, right-click on the folder to access the context menu.

2. In the context menu, click **New > JSON Document**.
   A selection of available templates is displayed.



3. Select a blank template or an existing template.
4. In **File Name,** enter the file name.
5. To create the JSON document, click **Finish**.
   The JSON document is created and opened in the JSON Editor.

3.3.3.1 JSON Editor

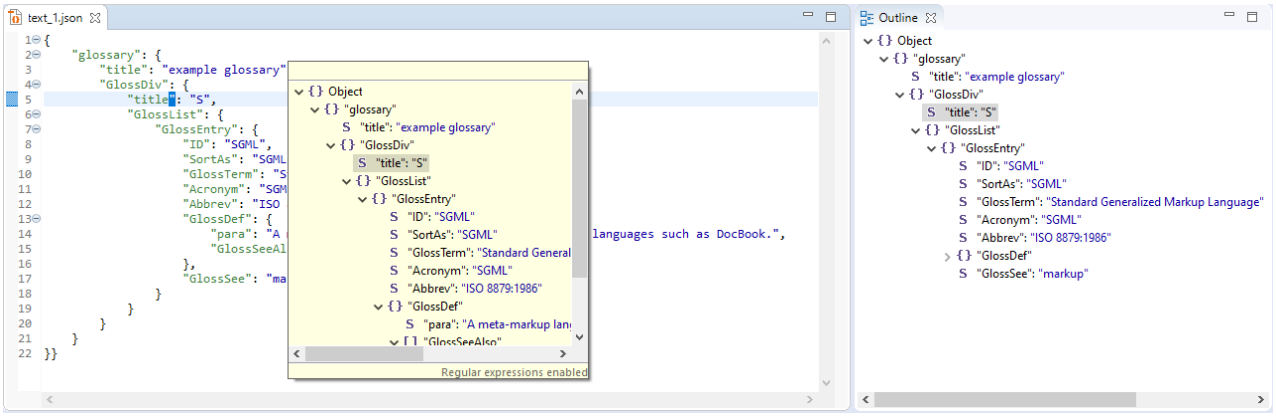
The JSON Editor allows JSON documents to be edited directly within the X4 Designer.

> To open the JSON Editor, either create a new JSON Document or open an already existing document.

The Outline view displays the structure of the opened JSON document. Moreover, you can also use the context menu **Quick Outline** or the shortcut `CTRL+O` to open the Quick Outline view and to search for keys and values within the JSON document.

Shortcuts

`CTRL+O`	Open the Quick Outline view
`CTRL+Shift+F`	Format the JSON code



The formatting of the JSON code and the colors for syntax highlighting can be changed under **Tools > Options > JSON Editor**. For more information, see Configuring the JSON Editor.

3.3.4 Document

`Document`: Includes any non-XML data, for example ASCII text or binary formats, in the process. It is tried to determine the MIME type from the file name extension. The system character encoding is used as character encoding.

If character encoding and MIME type are not determined correctly, then use the Document Properties Changer to set them accordingly.

Properties

`Document`	Path to the linked text or binary file in the repository.
`Label`	Label of the symbol within the process diagram
`Operation`	Determines if the file is read or written **Possible values:** • *`Read`*: read file into the process and pass data to the next process step (default) • *`Write`*: save file with the data from the previous process step
`Type`	Defines the type of the file **Possible values:** • `Resource` • `Buffer` (default)

3.3.4.1 ASCII Descriptor

An (ASCII) descriptor is an XML document describing the structure of data records in a text file. It serves the adapters ASCII Writer, COBOL Copybook Writer and COBOL Copybook Reader as rules definition for the conversion between text and XML. The repository path of descriptor documents is specified in this adapters as parameter. Thus, descriptors normally are not included as own process component into processes.

> For text files in *Cobol Copybook* format, you can generate a specific descriptor document using the Cobol Copybook to XML Wizard automatically. For CSV text documents, you can use the CSV Converter.

Elements

`<Descriptor>`	Root element. Contains one or multiple elements `<Record>`, as well as attributes described below.
`<Record>`	Corresponds to a data record in the text file. It can contain one or multiple elements `<Column>` or `<Record>`, as well as attributes described below.
`<Column>`	Corresponds to a column, i.e., a concrete value in the text file, and has attributes described below, but it may not have child elements.

Attributes of <Descriptor>

Attribute	Description
`Id`	Data record ID (optional) > The attribute is used within the ASCII Reader and ASCII Writer.
`Name`	Name of the root element in the XML document > The attribute is used within the GDV Message Writer, the COBOL Copybook Writer (`String`) and the COBOL Copybook Reader (`String`).

Attribute	Description
`Description`	Description text for the ASCII descriptor (optional)

`RecordSeparator`	Separator for the data records (optional)
	
	• The attribute is used within the GDV Message Writer, the ASCII Reader (`EscapedString`) and ASCII Writer (`EscapedString`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`).
	• In case of `EscapedString` some escape characters are replaced automatically:
	  • `\n`: new line (`newline`)
	  • `\r`: carriage return (`carriage return`)
	  • `\t`: Tab – move the cursor forward by a specified space (`tab`)
	  • `\u....`: hexadecimal numbers with 4 digits are converted to integer, e.g., `\u00FF` to *255*

`Columns`	Required number of columns of the data record (optional) The attribute is used within the ASCII Reader (`integer`) and ASCII Writer (`integer`).
`RecordOrder`	Order of the single data records (optional) The attribute is used within the ASCII Reader and ASCII Writer. **Possible values:** • `Fix`: the order remains as in text file • `Loose`: the order from the ASCII descriptor is adopted
`RecordLength`	Length of the single data records (optional) The attribute is used within the GDV Message Reader (`integer`).
`Identifier`	Value of the data record identifier (optional) The attribute is used within the ASCII Reader (`String`) and ASCII Writer (`String`).

Attribute	Description
`IdentifierPos`	Position of the data record identifier (optional) ⓘ The attribute is used within the GDV Message Reader (`integer`), the ASCII Reader (`integer`) and ASCII Writer (`integer`) and the COBOL Copybook Writer (`integer`) and COBOL Copybook Reader (`integer`).
`IdentifierLength`	Lenght of the data record identifier (optional) ⓘ The attribute is used within the GDV Message Reader (`integer`), the ASCII Reader (`integer`) and ASCII Writer (`integer`) and the COBOL Copybook Writer (`integer`) and COBOL Copybook Reader (`integer`).
`TeilsatzPos`	Specifies the position within the record data ⓘ The attribute is used within the GDV Message Reader (`integer`). **Possible values:** • `Identifier = TeilsatzPos`: sub-records are read • `0`: all sub-records are read

`Suffix`	Suffix (optional). Marks the end of a data record.
	
	ⓘ • The attribute is used within the ASCII Reader (`EscapedString`) and ASCII Writer (`EscapedString`).
	• In case of `EscapedString` some escape characters are replaced automatically:
	  • `\n`: new line (`newline`)
	  • `\r`: carriage return (`carriage return`)
	  • `\t`: Tab – move the cursor forward by a specified space (`tab`)
	  • `\u....`: hexadecimal numbers with 4 digits are converted to integer, e.g., `\u00FF` to *255*

Attribute	Description

`Prefix`	Prefix (optional). Marks the beginning of a data record
	
	• The attribute is used within the ASCII Reader (`EscapedString`) and ASCII Writer (`EscapedString`).
	• In case of `EscapedString` some escape characters are replaced automatically:
	  • `\n`: new line (`newline`)
	  • `\r`: carriage return (`carriage return`)
	  • `\t`: Tab – move the cursor forward by a specified space (`tab`)
	  • `\u....`: hexadecimal numbers with 4 digits are converted to integer, e.g., `\u00FF` to *255*

`Mandatory`	Declare data record as mandatory (optional) The attribute is used within the ASCII Reader and ASCII Writer. **Possible values:** • *`true`*: mandatory data record • *`false`*: no mandatory data record
`Multiple`	Data record can exist multiple times (optional) The attribute is used within the ASCII Reader and ASCII Writer. **Possible values:** • *`true`*: Data record can exist multiple times (see attribute `Occurs`) The attribute `Occurs` must contain a value, otherwise the error *`java.lang.NumberFormatException: null`* will be output. • *`false`*: Data record can exist only once.
`Length`	Required length of the data record in characters (optional) The attribute is used within the ASCII Reader (`LengthFormat`) and ASCII Writer (`LengthFormat`).

Attribute	Description

`Separator`	Separator for the data records (optional)
	
	• The attribute is used within the ASCII Reader (`EscapedString`) and ASCII Writer (`EscapedString`).
	• In case of `EscapedString` some escape characters are replaced automatically:
	  • `\n`: new line (`newline`)
	  • `\r`: carriage return (`carriage return`)
	  • `\t`: Tab – move the cursor forward by a specified space (`tab`)
	  • `\u....`: hexadecimal numbers with 4 digits are converted to integer, e.g., `\u00FF` to *255*

Attributes of <Record>

Attribute	Description
`Id`	ID of the data record The attribute is used within the ASCII Reader (`String`) and ASCII Writer (`String`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`).
`Name`	Name of the data record elements in the XML document The attribute is used within the GDV Message Reader and GDV Message Writer, the ASCII Reader (`String`) and ASCII Writer (`String`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`).

Attribute	Description

`Prefix`	Prefix (optional). Marks the beginning of a data record.
	
	• The attribute is used within the ASCII Reader (`EscapedString`) and ASCII Writer (`EscapedString`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`).
	• In case of `EscapedString` some escape characters are replaced automatically:
	  • `\n`: new line (`newline`)
	  • `\r`: carriage return (`carriage return`)
	  • `\t`: Tab – move the cursor forward by a specified space (`tab`)
	  • `\u....`: hexadecimal numbers with 4 digits are converted to integer, e.g., `\u00FF` to *255*

`Suffix`	Suffix (optional). Marks the end of a data record.
	
	• The attribute is used within the ASCII Reader (`EscapedString`) and ASCII Writer (`EscapedString`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`).
	• In case of `EscapedString` some escape characters are replaced automatically:
	  • `\n`: new line (`newline`)
	  • `\r`: carriage return (`carriage return`)
	  • `\t`: Tab – move the cursor forward by a specified space (`tab`)
	  • `\u....`: hexadecimal numbers with 4 digits are converted to integer, e.g., `\u00FF` to *255*

`Mandatory`	Declare data record as mandatory (optional) The attribute is used within the ASCII Reader and ASCII Writer and the COBOL Copybook Writer and COBOL Copybook Reader. **Possible values:** • *`true`*: mandatory data record • *`false`*: no mandatory data record

Attribute	Description
`Multiple`	Data record can exist multiple times (optional) ⓘ The attribute is used within the ASCII Reader and ASCII Writer and the COBOL Copybook Writer and COBOL Copybook Reader. **Possible values:** • *`true`*: Data record can exist multiple times (see attribute `Occurs`) ⓘ The attribute `Occurs` must contain a value, otherwise the error *`java.lang.NumberFormatException: null`* will be output. • *`false`*: Data record can exist only once.
`IdentifierPos`	Position of the data record identifier (optional) ⓘ The attribute is used within the ASCII Reader (`integer`) and ASCII Writer (`integer`).
`IdentifierLength`	Length of the data record identifier (optional) ⓘ The attribute is used within the ASCII Reader (`integer`) and ASCII Writer (`integer`).
`Identifier`	Value of the data record identifier (optional) ⓘ The attribute is used within the GDV Message Writer and GDV Message Reader, the ASCII Reader (`String`) and ASCII Writer (`String`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`).

Attribute	Description

`ColumnSeparator`	Column separator (optional)
	
	• The attribute is used within the ASCII Reader (`EscapedString`) and ASCII Writer (`EscapedString`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`).
	• In case of `EscapedString` some escape characters are replaced automatically:
	  • `\n`: new line (`newline`)
	  • `\r`: carriage return (`carriage return`)
	  • `\t`: Tab – move the cursor forward by a specified space (`tab`)
	  • `\u....`: hexadecimal numbers with 4 digits are converted to integer, e.g., `\u00FF` to *255*

`Columns`	Number of required columns of the data record (optional)
`ColumnOrder`	Order of the single columns (optional) The attribute is used within the ASCII Reader and ASCII Writer. **Possible values:** • `Fix`: the order remains as in text file • `Loose`: the order from the ASCII descriptor is adopted
`Occurs`	Number of repetitions of the data record The attribute is used within the ASCII Reader (`integer`) and ASCII Writer (`integer`) and the COBOL Copybook Writer (`integer`) and COBOL Copybook Reader (`integer`).
`Description`	Description text for the data record (optional)
`Length`	Required length of the data record in characters (optional) The attribute is used within the GDV Message Reader (`integer`), the ASCII Reader (`integer`) and ASCII Writer (`integer`) and the COBOL Copybook Writer (`integer`) and COBOL Copybook Reader (`integer`).

Attributes of <Column>

Attribute	Description
`Id`	Column ID (optional) ⓘ The attribute is used within the ASCII Reader (`String`) and ASCII Writer (`String`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`).
`Name`	Name of the field element within the XML document ⓘ The attribute is used within the GDV Message Reader (`String`) , the ASCII Reader (`String`) and ASCII Writer (`String`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`).
`Type`	Data type of the column/field ⓘ The attribute is used within the GDV Message Writer (`Integer` | `String`), the ASCII Reader (`Integer` | `String`) and ASCII Writer (`Integer` | `String`) and the COBOL Copybook Writer (`Integer` | `String`) and COBOL Copybook Reader (`Integer` | `String`). **Possible values:** • *`String`*: string value • *`Number`*: numeric value

`Prefix`	Prefix (optional). Marks the beginning of a field value.
	
	ⓘ • The attribute is used within the ASCII Reader (`EscapedString`) and ASCII Writer (`EscapedString`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`).
	• In case of `EscapedString` some escape characters are replaced automatically:
	  • `\n`: new line (`newline`)
	  • `\r`: carriage return (`carriage return`)
	  • `\t`: Tab – move the cursor forward by a specified space (`tab`)
	  • `\u....`: hexadecimal numbers with 4 digits are converted to integer, e.g., `\u00FF` to *255*

Attribute	Description

`Suffix`	Suffix (optional). Marks the end of a field value.
	
	• The attribute is used within the ASCII Reader (`EscapedString`) and ASCII Writer (`EscapedString`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`).
	• In case of `EscapedString` some escape characters are replaced automatically:
	  • `\n`: new line (`newline`)
	  • `\r`: carriage return (`carriage return`)
	  • `\t`: Tab – move the cursor forward by a specified space (`tab`)
	  • `\u....`: hexadecimal numbers with 4 digits are converted to integer, e.g., `\u00FF` to *255*

`Mandatory`	Declare as mandatory field (optional) The attribute is used within the ASCII Reader and ASCII Writer and the COBOL Copybook Writer and COBOL Copybook Reader. **Possible values:** • *`true`*: mandatory field • *`false`*: no mandatory field
`Multiple`	Field can exist multiple times (optional) The attribute is used within the ASCII Reader and ASCII Writer. **Possible values:** • *`true`*: Field can exist multiple times • *`false`*: Field can exist only once
`IdentifierPos`	Position of the column identifier (optional) The attribute is used within the ASCII Reader (`integer`) and ASCII Writer (`integer`).

Attribute	Description
`IdentifierLength`	Length of the column identifier (optional) ⓘ The attribute is used within the ASCII Reader (`integer`) and ASCII Writer (`integer`).
`Identifier`	Value of the column identifier (optional) ⓘ The attribute is used within the ASCII Reader (`integer`) and ASCII Writer (`integer`).
`Default`	Default value of the field (only for mandatory fields) ⓘ The attribute is used within the ASCII Reader (`String`) and ASCII Writer (`String`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`).
`Regex`	Regular expression describing the column values (optional) ⓘ The attribute is used within the ASCII Reader (`String`) and ASCII Writer (`String`) and the COBOL Copybook Writer (`String`) and COBOL Copybook Reader (`String`). *Example*: *`*[0-9a-fA-F]+.*`*

`Length`	Required field length in characters (optional)
	
	ⓘ
	• The attribute is used within the GDV Message Writer (`integer`) and GDV Message Reader (`integer`), the ASCII Reader (`LengthFormat`) and ASCII Writer (`LengthFormat`) and the COBOL Copybook Writer (`LengthFormat`) and COBOL Copybook Reader (`LengthFormat`).
	• Within COBOL Copybook Writer and COBOL Copybook Reader the following values are possible:
	  • `integers`: e.g., 10 is interpreted as 10.0
	  • `no value`: is interpreted as 0.0
	  • `decimals`: e.g., 10,2 is interpreted as 10.2

Attribute	Description
`RecordMultiplicator`	Repeat data record ⓘ The attribute is used within the COBOL Copybook Writer and COBOL Copybook Reader. **Possible values:** • *true*: repeat field (specified in `RecordMultiplicated`) • *false*: do not repeat field
`RecordMultiplicated`	ID of the multiple occurring data record (if `RecordMultiplicator` is set to *True*) ⓘ The attribute is used within the COBOL Copybook Writer (`integer`) and COBOL Copybook Reader (`integer`).
`Binary`	When reading/writing the data is encoded from Base64 to binary or from binary to Base64 ⓘ The attribute is used within the COBOL Copybook Writer and COBOL Copybook Reader.
`Occurs`	Number of repetitions of the column ⓘ The attribute is used within the COBOL Copybook Writer (`integer`) and COBOL Copybook Reader (`integer`).

3.3.5 Subprocess

`Subprocess`: Provides the possibility to outsource parts of processes into separate process files. You can include each process in another process as a subprocess. In doing so, a subprocess can receive an input from the preceding process module and passes on the last status within the subprocess as well as an output document, if applicable, to the subsequent process module in the parent process.

ⓘ
- To edit the subprocess, double-click the corresponding icon in the drawing area.
- If an input or output schema was stored when the process was created, it can be called up via the context menu entry **Show input schema** or **Show output schema**.

Properties

`Document`	Relative path to the subprocess ℹ This path is dynamically parameterizable via the `workflow` parameter. The use of placeholders (e.g., `%START_PROCESS_PATH%` or `%CALLING_PROCESS_PATH%`) is also possible (not in debug mode of the X4 Designer)
`Label`	Label of the symbol within the process diagram

Creating subprocesses

For more extensive processes, it is useful to outsource related modules as subprocesses. This makes the process clearer and the subprocesses can be used more than once. Processes can be nested into each other to practically any depth. Subprocesses are saved as separate `.wrf` files.

> Like other process modules, processes can be dragged and dropped from the **Projects** view into the drawing area of the Process Editor.

1. Mark at least two modules, including all transition lines that are to be outsourced.
2. Select the **Pack As Subprocess** function from the context menu:
   - In **File name**, enter the name of the subprocess.
   - Click **Finish**.
3. The subprocess with the selected modules is created as an independent process file in the repository. In the main process, the selected modules are automatically replaced by the newly created subprocess module.

3.3.5.1 Using private subprocesses

Subprocesses can be used privately. Private subprocesses can only be reused for their parent process. A private subprocess can either be newly created or packed from existing process modules.

To **create a private subprocess**, follow these steps:

1. Select an existing process in the **Projects** view.
2. In the menu, click on **Project** > **Add Process...**.

   > **Alternative**
   > Right-click on an existing process. In the context menu, click **New** > **Process**.

   A new process is created as a private subprocess of the existing process and can be edited.

To **pack a private subprocess** from existing process modules, follow these steps:

1. Open an existing process.
2. Mark at least two modules, including all transition lines.
3. Right-click on the selected modules.
4. Click **Pack as Private Subprocess** in the context menu.
5. In **File name,** enter the name of the subprocess.

6. Confirm with **Finish**.
   The marked modules are created as a private subprocess and can be edited there.

3.3.6 Mapping

`Mapping`: Transforms an XML document in a source structure via an XSLT stylesheet into a target structure (usually XML) and thus enables the individual data elements to be assigned to one another (so-called *Mapping*). XSLT also serves X4 BPMS as a scripting language that can be used to dynamically parameterize process modules.

Properties

`Document`	Path to the linked file in the repository
`Label`	Label of the symbol within the process diagram
`Component Parameters`	Set component parameters. For more information, see Component Parameter.
`Empty Input` *(X4EMPTYINPUT)*	Do not process data from the previous process step in the XSL mapping **Possible values:** • *`true`*: do not process input data ⓘ If `Empty Input` is set to `true`, the `Mapping` component is additionally marked. • *`false`*: process input data (default)
`Skip Dynamic Params` *(X4SKIPDYNPARAM)*	Disable evaluation of dynamic parameters. This can significantly speed up the execution of XSL mappings. **Possible values:** • *`true`*: disable evaluation of dynamic parameters ⓘ If `Skip Dynamic Params` is set to `true`, the `Mapping` component is additionally marked. • *`false`*: process dynamic parameters in XSL mapping (default)

`Remove X4 Processing Instructions`	Remove X4 Processing Instructions in the mapping result. **Possible values:** • *`true`*: delete X4 Processing Instructions from the result (default) • *`false`*: display X4 Processing Instructions in the result ⓘ If `Remove X4 Processing Instructions` is set to `false`, the `Mapping` component is additionally marked.

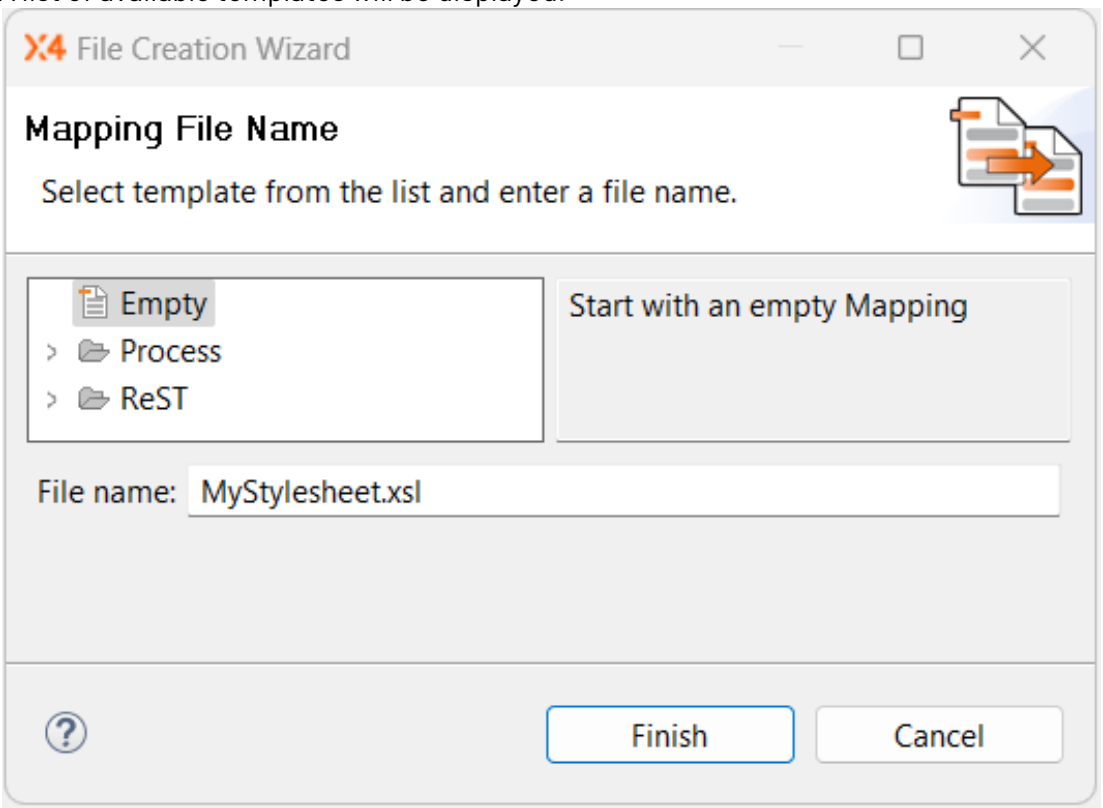
3.3.6.1 Creating and Using an XSL Mapping

Mappings can be created within an ESB project in the **Transformations** folder. There are two options available for this purpose:

- via the **Project** > **Add XSL Stylesheet** menu
- via the **New** > **XSL Stylesheet** context menu

To create an XSL mapping:

1. Right-click on the **Transformations** folder within the ESB project to open the context menu.
2. In the context menu, select **New** > **XSL Stylesheet**.
   A list of available templates will be displayed.



3. Select an empty or an already available template.
4. Enter a file name in **File Name**.

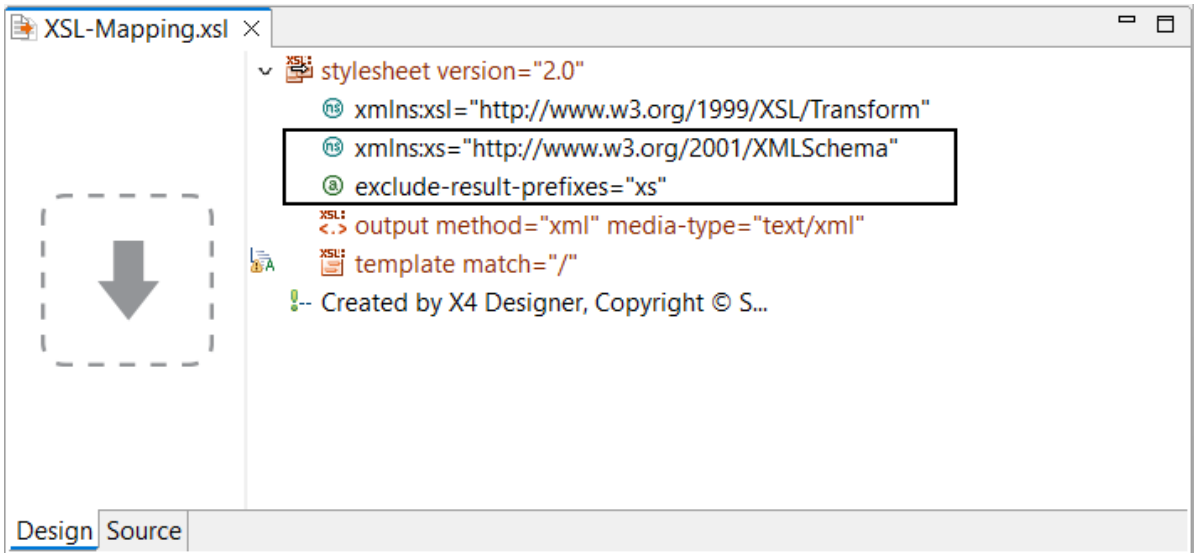
5. Select **Finish** to create the XSL mapping.
   The XSL mapping has been created and will be opened in the Mapping Editor.

Defining the source and target structure in the XSL mapping

In the first step, you define the source data structure of the transformation. That is, you fill the left pane of the XSL mapping with source data.

There are two ways to do this:

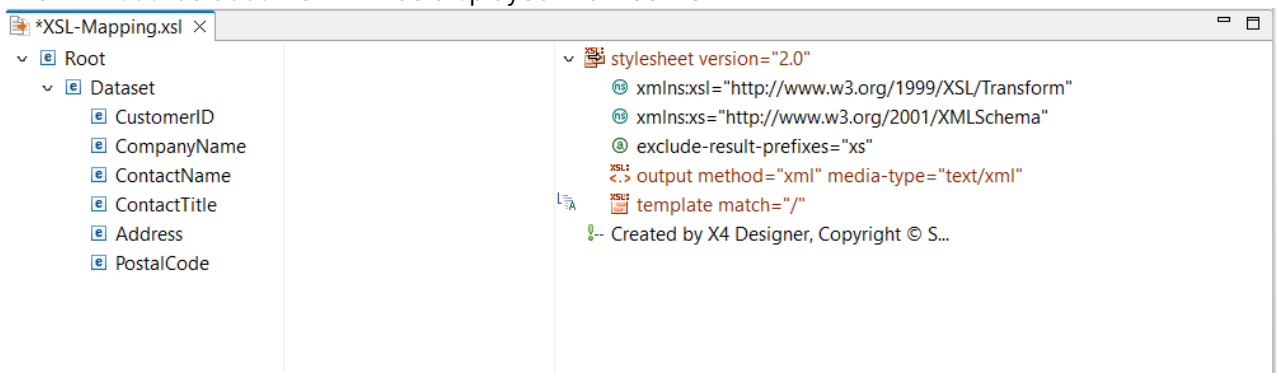
1. Insert the XML source document (XML document or XML schema) from the **Projects** view via drag-and-drop into the Mapping Editor's source pane.
   Alternatively, you can right-click in the source document area of the XSL mapping and select **Assign Input** to choose a file to insert. All projects from the **Projects** view are available for selection.



> The namespace `xmlns:xs="http://www.w3.org/2001/XMLSchema"` and the `exclude-result-prefixes="xs"` attribute allow the use of data types in conjunction with XPath 2.0. Both entries are only included in newly created XSL stylesheets that are not based on a template.

2. Right-click in the source document area of the XSL mapping and select **Paste Input** to paste the content from the clipboard.
   Example: When debugging a process, you can copy the data from the **Output** view to the clipboard and paste it in the XSL mapping, or you can copy XML data from an editor and paste it in the mapping.

The XML source document will be displayed in a tree view.



In the next step, you create the basic structure of the target document in the XSL mapping. To do this, proceed as follows:

1. In the stylesheet pane, right-click on the root template (`template match="/"`) to open the context menu.
2. In the context menu, select **XSL Element** to open the **XSL Element Creation Wizard**.
3. Insert all required elements in the correct hierarchy.
   Example: The structure of the target document is an HTML skeleton (`<html>`, `<body>`) that contains a table (`<table>`) with a row (`<tr>`) having three header cells (`<th>`) and another row (`<tr>`) having three data cells (`<td>`).

   > If you already know the XML structure to be generated by the XSL mapping, you can assign an XML document or XML schema from the Projects view to the stylesheet pane via drag-and-drop. This structure can be used as target structure for your mapping. When assigning an XML structure via drag-and-drop, a dialog will come up to ask how it shall be inserted:
   > - **Full XML structure including data** inserts the complete XML document structure, including its element values and attribute values.
   > - **Virtual nodes (structure without data)** inserts only the tree structure as virtual nodes into the stylesheet pane. The virtual nodes are created in the target structure only when double-clicking them or by defining a data mapping.

4. If you want to add text, select **Add Other Nodes > Text** from the context menu.
   Example: Insert a column label text into the table header cell elements.

Using data mapping: *for-each*

Use `<xsl:for-each>` to generate a target structure that repeats for each data row:

1. In the source document, select the node structure. Hold down the `Shift` key, and assign the structure via drag-and-drop to the target element to be repeated.
   A `for-each` element will be created.

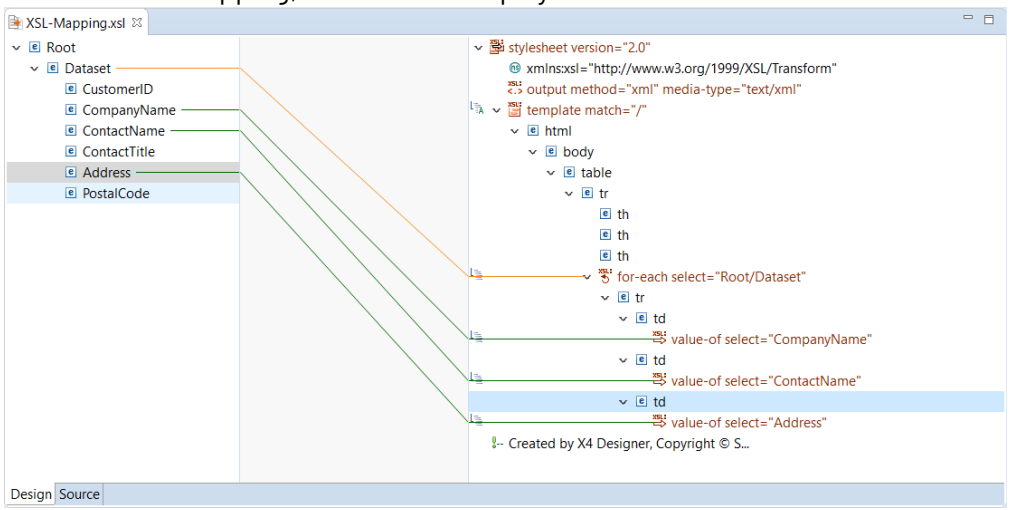
Example: Assign `<Dataset>` to the second `<tr>` element in order to generate HTML table rows dynamically.

2. Ensure that the `for-each` element is an immediate parent node of the second `<tr>` element.

Using data mapping: *value-of*

Use `<xsl:value-of>` to assign node contents from the source document to the target document:

1. Click the source node and connect it with the target node via drag-and-drop.
   Example: Connect the `<CustomerName>` element with the first `<td>` element, the `<CompanyName>` element with the second `<td>` element, and the `<Address>` element with the third `<td>` element.
2. Drag further mapping lines.
   For each data mapping, a line will be displayed.



Using data mapping: *copy-of*

Use `<xsl:copy-of>` to copy a node including all child nodes (elements, attributes, namespace definitions, etc.):

1. In the source document, select the node to be copied.
2. Hold down the `Ctrl` key and assign the node via drag-and-drop to the target node.

A `copy-of` element will be created.

Source view: Using XSLT functions

> **Example**
> You want to merge the contents of the `ContactName` and `ContactTitle` elements using the `concat` function.

1. Click the **Source** tab to display the XSL mapping in the source view.

2. Edit the XSLT syntax.
   Example: Replace the line `<xsl:value-of select="ContactName"></xsl:value-of>` by `<xsl:value-of select="concat(ContactName, ', ',ContactTitle)"/>`.

Changing the Output Format

1. In element `<xsl:output>`, modify the output format in attribute `method="xml"`.
   Example: To generate an HTML 4.0 document, set `method="html"`.
2. In element `<xsl:output>`, modify the output document's MIME type defined in attribute `media-type="text/xml"`.
   Example: To generate an HTML 4.0 document, set `mime-type="text/html"`.

Testing the XSL Mapping

If an input file is available, you can use the *Mapping Editor's* integrated preview function.

1. Save the mapping via `Ctrl+S`.
2. In the toolbar, click ▶ to test the XSL mapping locally.
   The transformation result is displayed in the Output view.

Using XSL Mappings in Technical Processes

1. Drag the XSL mapping file from the **Projects** view into an opened technical process via drag-and-drop.
   Example: Insert it between the XML document and the `Stop` component.
2. If necessary, define parameters within the property `Component Parameters`.
   These will be passed to the XSL mapping during the process execution, see Parameterizing Process Components.

3.3.6.2 Mapping Editor

The Mapping Editor allows to create and edit XSL mappings in a graphical interface. Using XSLT, you can transform an XML data structure into another XML data structure or into an XSL-FO document. The Mapping Editor supports you with a graphical interface and with an automatic stylesheet and XPath validation when creating or editing XSL mappings.

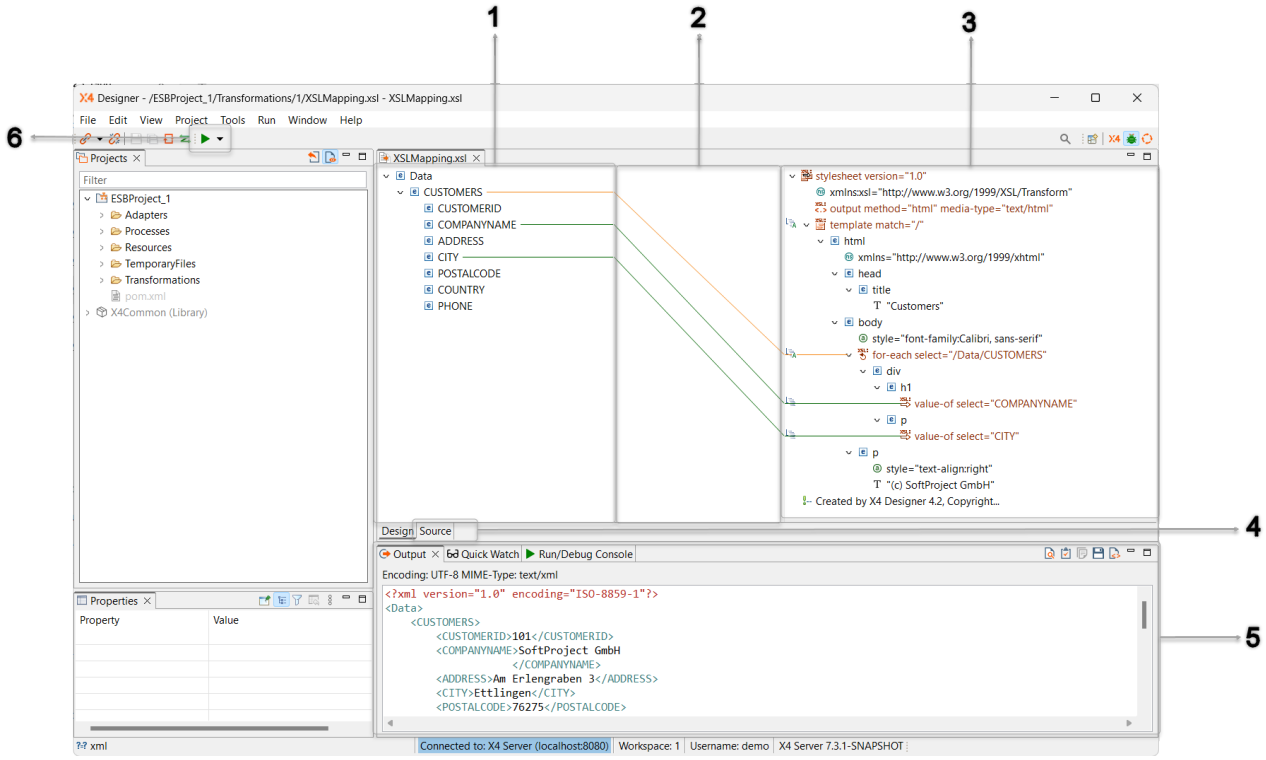
An XSL mapping involves three documents:

- XML source document (or an XML schema to provide the input structure)
- XSLT stylesheet (created with the *Mapping Editor*)
- Result document (in most cases: XML)

> ⓘ To open the Mapping Editor, create a new mapping or open an already existing XSL mapping.

The Mapping Editor's structure

The Mapping Editor is divided into the following areas:



1	**Source pane**	The source document to be transformed or an XML Schema file (`.xsd`) is displayed here. Using the context menu, you can edit selected nodes as follows: • **Copy XPath for this element to Clipboard**: copy XPath expression of the selected element to the clipboard and use it elsewhere • **Assign Input**: assign input XML document • **Remove Input**: remove assigned input XML document • **Paste Input**: insert XML document from the clipboard
2	**Linking pane**	The Linking pane displays lines that visualize the relationships between source and stylesheet nodes. ⓘ The chapter Tastenkombinationen im Mapping Editor gives you an overview of the most important key combinations helping you to work with XSL stylesheets.

3	**Stylesheet pane**	The Stylesheet pane is where you edit the stylesheet. It displays the XSL stylesheet in an editable tree with specialized features for editing XSLT stylesheets.
		
		The different parts of the stylesheet are displayed with symbols and colored highlighting in the tree view:
		
		• Nodes that contain XSLT instructions are colored brown and have a corresponding icon, e.g.,
		• If a node contains an XPath expression, it is indicated by the XPath icon on the left side. You can click on this icon to edit the XPath expression.
		• Virtual nodes (incurred nodes from an assigned XML document or XML schema that do not yet explicitly exist in the XSLT stylesheet) are greyed out.
		
		Using the context menu in the stylesheet pane, you can edit selected nodes as follows:
		
		• **Edit XSL Element**: Edit selected XSL element
		• **Rename**: Rename the node
		• **Element**: Insert an element
		• **Attribute**: Insert attribute
		• **Edit**: (In case of processing instructions and namespace declarations) Edit node content
		• **XSL Element**: Insert XSL element
		  • `as Child`: as child node
		  • `Before:` as node before the selected node
		  • `After:` as node after the selected node
		• **Add other nodes**: Add other nodes (`Namespace`, `Text`, `Comment`, `Processing Instruction`)
		• **Add here**: (In case of virtual nodes) Convert a virtual node into a real node
		• **Add Processing Instructions**: Insert processing instructions to set Dynamic Parameters.
		  • `Condition State` for status information
		  • `Log Info` for log messages
		  • `Dynamic Parameter` to set dynamic values
		  • `No Result` for empty mapping outputs
		• **Add Parameter**: Insert system, global or project parameters, see Creating and Using Parameters
		• **Add Parent**: Insert a parent element (superordinate element)
		• **Cut/Copy/Paste**: Cut, copy, or paste selection
		• **Delete:** Delete the selected node
		• **Expand/Collapse**: Expand or collapse the child nodes of the selected node
		• **Assign Virtual Nodes**: Insert virtual nodes from an XML document or schema
		• **Remove Virtual Node**: Delete virtual nodes from an XML document or schema

		The **XSL Element Creation Wizard** allows to create XSL mappings quickly and easily. The wizard offers the complete range of XSLT 2.0 elements and enables to create them easily via the context menu. Within the wizard, you can specify whether the new element is to be created before, after or as a child of the selected element. Features like auto-completion, or the display of all available and allowed XSL elements enable faster work. Moreover, the wizard allows to set and to edit the attributes of the XSL element. Required attributes are set automatically, optional attributes can be selected and filled as needed. For attributes that have multiple options to choose from, a drop-down list with all available values is provided.
4	**Source and Design view**	You can switch between two different views within the Mapping Editor: • **Design** : The editor displays the XSLT stylesheet as a tree in a three column view. Here, you can graphically map nodes of the source document/schema to the XSLT stylesheet nodes by drawing connection lines. • **Source** : In this source code view, you can edit the XSLT stylesheet directly in an XML editor
5	**Output**	The result of locally run XSL mappings is displayed in the Output view. The following additional actions can be performed via the toolbar: • *Save content of output view*: Save the data form the output view • *Format the content of the output view*: Format the data from the output view
6	**Run transformation**	Via **Run Transformation,** you can run the XSL mapping locally and display the transformation result in the output view (5).

3.3.7 PDF Creator

`PDF Creator`: Transforms an input XML document by means of an XSLT stylesheet into an XSL-FO document. In the same process action, this document is converted into a page-oriented output document using the XSL-FO processor *Apache FOP 1.0*. Possible output formats for Reports are PDF, PDF/A, and PostScript.

Properties

`Document`	Path to the linked file in the repository
`Component Parameters`	Set component parameters, see Using Component Parameters
`Report type`	Output format of the report **Possible values:** • *PDF*: PDF document (Portable Document Format) • *PDF/A*: PDF document in the archiving format *PDF/A-1b* according to *ISO 19005-1* • *PS*: PostScript file ⓘ Using the *PDF/A* option, Apache FOP 1.0 requires specifying a font (using attribute `font-family`) for each text block of within the FO document. In addition, all used fonts must be available on the system during PDF/A generation.

Creating reports with XSL-FO

The `PDF Creator` component allows to dynamically generate documents for print (reports) during process runtime. A report is a specific XSL mapping that transforms input XML data and outputs an XML document with XSL-FO syntax. The X4 BPMS will convert the generated XSL-FO document automatically into the specified output format and will hand it over to the next process step.

The X4 BPMS uses Apache FOP as XSL-FO processor to process reports.

The XSL-FO element `<fo:external-graphic src="url('path/to/file.png')">` allows to embed graphic files (e.g., GIF, JPEG, PNG, SVG) into your report document. You can also specify the graphics path using `xstore:/` URLs to reference files from the X4 repository, e.g.:

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

The optional attributes `content-width` and `content-height` specify the graphic's width and height in the output document.

As an alternative, you can embed binary graphic files into the report as Base64-encoded string, see also Base64 Converter:

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

The source code of SVG images can also be embedded directly into the report, e.g.:

```
<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>
```

Note that all SVG elements must use the namespace `svg:`.

Example

The following example is a very simple XSL-FO document:

```
<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>
```

For more information on **hyphenation**, visit https://xmlgraphics.apache.org/fop/2.5/hyphenation.html.

3.4 Drawing Elements

The following drawing elements can be used in Technical Processes:

3.4.1 Connector

`Connector`: Creates a logical connection between elements

Connector lines can only be used to visualize a logical coherence!
When executing the Business Process via the X4 Server, only process components are interpreted as data and control flow elements that have been connected with Sequence Flows in Business Processes, and those components that are connected with Transitions in Technical Processes.

Properties

`Label`	Free text area to label the connector line (useful to layout/ document the process diagram)
`Line Routing`	Line routing **Possible values:** • *`Angled`*: Angled line • *`Straight`*: Straight line (Default)
`Line Type`	Line style **Possible values:** • *`Dash line`*: Dashed line (Default) • *`Solid line`*: Solid line • *`Dot line`*: Dotted line
`Line Color`	Connector line color. To change the color, select the property and click . **Possible values:** • *`RGB(RRR,GGG,BBB)`*: 24-bit RGB color code (can be selected with a color picker dialog) • *`RGB(0,0,0)`*: Black (Default)

`Line Source Decorator` and `Line Target Decorator`	Symbol for the beginning of the line **Possible values:** • *`Circle`*: Circle (Default) • *`Empty arrow`*: Empty arrow • *`Association arrow`*: Open arrow • *`Arrow`*: Filled arrow • *`Inverted arrow`*: Inverted arrow • *`Diamond`*: Filled rhombus • *`Empty diamond`*: Empty rhombus • *`Plain`*: No symbol

3.4.2 Rectangle and Ellipse

`Rectangle`	Draws a rectangle
`Ellipse`	Draws an ellipse

Properties

`Line Color`	Border color of the drawing element. To change the color, select the property and click . **Possible values:** • *`RGB(RRR,GGG,BBB)`*: 24-bit RGB color code (can be selected with a color picker dialog) • *`RGB(0,0,0)`*: Black (Default)
`Line Width`	Width of the drawing element's border **Possible values:** • any positive integer number • *`1`*: line with a width of 1 pixel (default)
`Hide Line`	Hide border of the drawing element

`Fill Color`	Fill color of the drawing element. To change the color, select the property and click . **Possible values:** • *`RGB(RRR,GGG,BBB)`*: 24-bit RGB color code (can be selected with a color picker dialog) • *`RGB(255,255,255)`*: White (default)
`Transparent`	Set drawing element as transparent. By enabling this property, the property `Fill Color` will be ignored.

3.4.3 Textbox

Abc `Textbox`: Inserts a text box, whose content can be edited and formatted in a dialog by double-clicking the box.

Properties

`Hyperlink`	Linking defined for the text box. To create a link, enter a valid URL. You can open the linked URL with `Ctrl` + `click`.
`Text`	Content of the text box. To edit and format the text, select the property and click .
`Rotation`	Rotation of the text box in degree **Possible values:** • *`0`*: no rotation (default) • *`90`*: rotate by 90° • *`180`*: rotate by 180° • *`270`*: rotate by 270°
`Fill Color`	Fill color of the text box. To change the color, select the property and click . **Possible values:** • *`RGB(RRR,GGG,BBB)`*: 24-bit RGB color code (can be selected via a color picker dialog) • *`RGB(255, 255, 255)`*: White (default)

`Transparent`	Set text box as transparent. By enabling this property, the property `Fill Color` will be ignored.

3.4.4 Polygon

`Polygon`: Draws a polygon with multiple corner marks

Properties

`Line Color`	Border color of the drawing element. To change the color, select the property and click . **Possible values:** • *`RGB(RRR,GGG,BBB)`*: 24-bit RGB color code (can be selected via a color picker dialog) • *`RGB(0,0,0)`*: Black (Default)
`Line Width`	Width of the drawing element's border **Possible values:** • any positive integer number • *`1`*: line with a width of 1 pixel (default)
`Fill Color`	Fill color of the drawing element. To change the color, select the property and click . **Possible values:** • *`RGB(RRR,GGG,BBB)`*: 24-bit RGB color code (can be selected with a color picker dialog) • *`RGB(255,255,255)`*: White (Default)
`Transparent`	Transparent fill (do not use a fill color); By enabling this property, the property `Fill Color` will be ignored.

3.4.5 Polyline and Line

`Polyline`	Draws a line with multiple way points
`Line`	Draws a straight line

Properties

`Line Color`	Border color of the drawing element. To change the color, select the property and click . **Possible values:** • *`RGB(RRR,GGG,BBB)`*: 24-bit RGB color code (can be selected with a color picker dialog) • *`RGB(0,0,0)`*: Black (default)
`Line Width`	Width of the drawing element's border **Possible values:** • any positive integer number • *`1`*: line with a width of 1 pixel (default)

3.4.6 Image

`Image`: Inserts an image frame that allows to include a `.gif`, `.jpg`, `.png`, or a `.bmp` image file.

> The inserted image will be stored within the process.

Properties

`Line Color`	Border color of the image frame. To change the color, select the property and click . **Possible values:** • *`RGB(RRR,GGG,BBB)`*: 24-bit RGB color code (can be selected with a color picker dialog) • *`RGB(0,0,0)`*: Black (default)
`Line Width`	Width of the drawing element's border **Possible values:** • any positive integer number • *`1`*: line with a width of 1 pixel (default)
`Image File`	Path to the linked image file

4 Working with ESB Projects

The X4 Repository is organized in projects. Projects have a file system structure and can contain any folders, processes, process components and other files.

4.1 Templates

4.2 The Structure of ESB Projects

ESB projects have a predefined and unchangeable folder structure, which is created automatically when a new ESB project is created.

> The automatically created folders and subfolders cannot be deleted, moved or renamed.

`Adapters`	Any number of adapters can be created here.
`Processes`	Any number of Technical Processes (`.wrf`) can be created here.
`Resources`	Folders, resources such as images, text and XML documents and other files can be stored here.
`TemporaryFiles`	Temporarily used files are stored here.
`Transformations`	Transformations (`.xsl`) and reports (`.rep`) are stored here.
`<Project>.wsinc`	Webservice definition, which is created and named after the project automatically when the project is created, e. g. *`ESBProject_Documentation.wsinc`*. > A project can contain only one webservice definition.

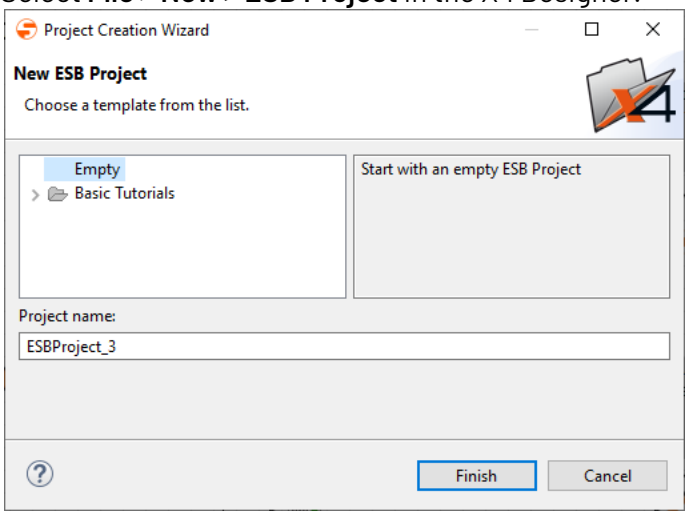
4.3 Creating ESB Projects

There are different options to create ESB projects:

- Via **New ESB Project** on the X4 Designer Welcome page
- Via the context menu **New** > **ESB Project** within the **Projects** view
- Via the **File** > **New** > **ESB Project** menu

How to create an ESB project

1. Select **File** > **New** > **ESB Project** in the X4 Designer.



2. Select an empty project template or an already existing template.

> It is also possible to create projects based on a template from an older version of the X4 Suite. These are migrated automatically during creation.

3. Enter the project name in **Project name**.
4. Click **Finish** to create the project.
   An empty ESB project with the predefined structure is created within the **Projects** view.
   You can now select and create various elements for your ESB project using the context menu. The available options change depending on where in the tree structure the context menu is opened.

> If you select the **Expert ESB Project** project type when creating the project, you can create an empty ESB project without a predefined folder structure. With this project type, you can work in expert mode and can freely define the folder structure and the elements it contains.
> **Notes on the usage of ESB projects from older X4 versions:**
>
> - ESB projects from X4 versions prior to 6.x will automatically be imported into an Expert ESB Project.
> - *Classic ESB Projects* from older X4 versions will be automatically transformed into an Expert ESB Project.

4.4 Managing Repository Projects

Learn how to manage projects in the Repository Navigator.

4.4.1 Import repository elements

Here you will learn how to import projects and various resources from the local file system into the repository.

4.4.1.1 Import via drag and drop

You can import one or more marked files into the Repository Navigator and thus into the currently connected repository by drag and drop from the file system (e.g. from the desktop or *Windows Explorer*). If a file with the same name already exists at the destination, you will be prompted to enter a new name.

> **Note:**
>
> - Folder structures are not supported during import, use the corresponding wizard for this.
> - For larger file transfers (over 1 MB in total), a confirmation dialogue will be displayed as there may be delays in the transfer as well as memory problems.
> - Imported files that contain special/space characters or umlauts in the file name must first be renamed before you can open them in X4 Designer.
> - Projects have a predefined structure so that the inserted files can only be stored in the respective predefined folder, see also The Structure of ESB Projects

4.4.1.2 Using the Import Wizard for files and projects

> **Note**
>
> - Projects have a predefined structure so that the files to be imported can only be stored in the respective predefined folder, see also The Structure of ESB Projects
> - Projects can only be imported at the top level and in the predefined project structure, see also The Structure of ESB Projects

1. Click on **Project > Import** in the menu.
   The **Resource Import Wizard** opens.
2. Select resource.
   - Select the file to be imported (`*.zip`) or the project to be imported via **Browse** and enter the corresponding name in **New project/file name**.
3. Click **Finish**.

4.4.2 Export repository elements

Here you will learn how to export projects and various resources from the repository to the file system.

4.4.2.1 Using the Export Wizard for files and projects

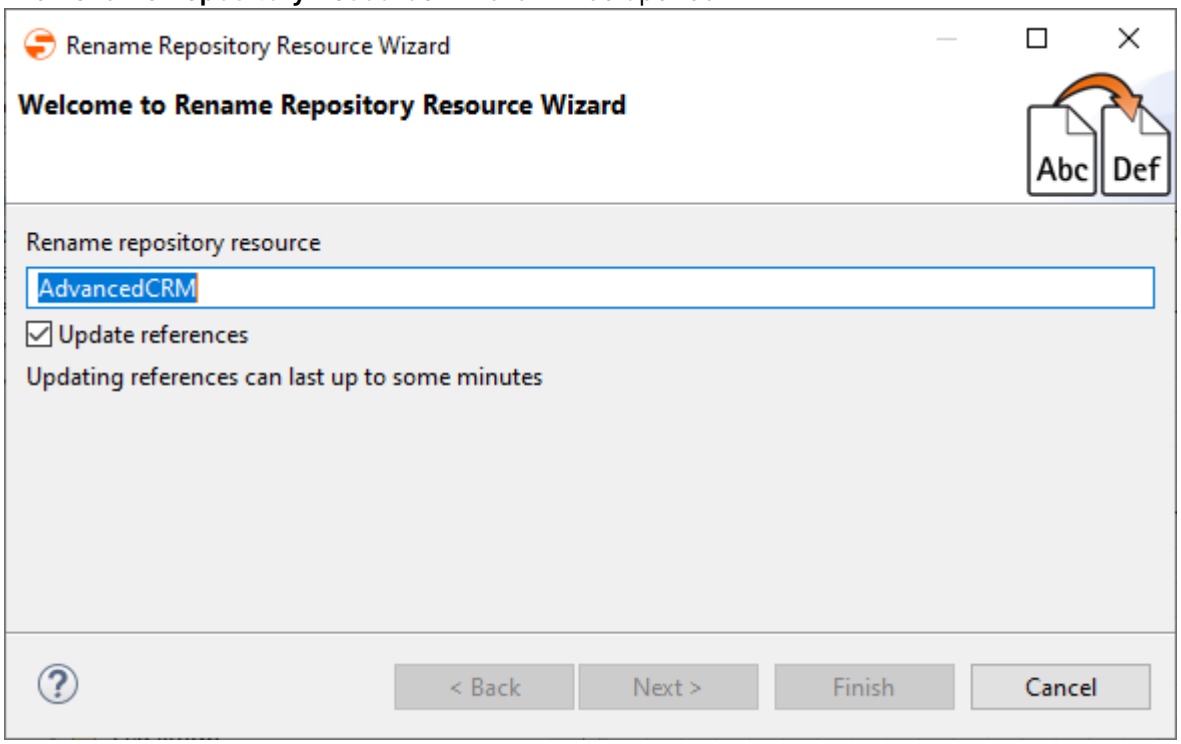
> **Note**
>
> - Projects can only be exported if the top project level is selected.

1. In the menu, click **Project > Export**.
   The **Resource Export Wizard** is opened.
2. Select destination folder.
3. Click **Save**.

4.4.3 Renaming elements and updating references automatically

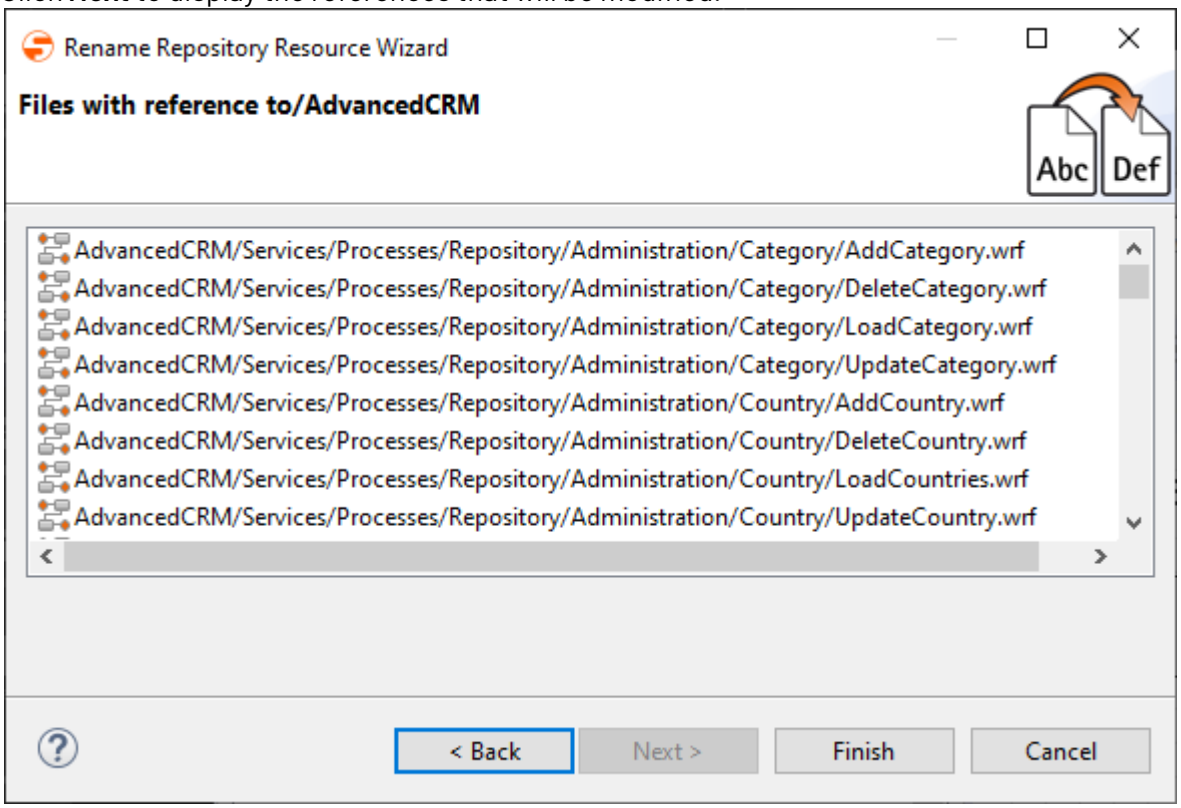
If you want to rename files in the repository, you can update references in processes automatically using the **Rename Repository Resource Wizard**.

1. Right-click on the corresponding element in the Repository Navigator to open the context menu.
2. Select **Refactor/Rename**.
   The **Rename Repository Resource Wizard** will be opened.



3. Enter a new name in **Rename repository resource**.

4. Click **Next** to display the references that will be modified.



5. Click **Finish** to apply the changes.
   The affected files are now renamed and the corresponding references to processes, resources or components are adjusted.

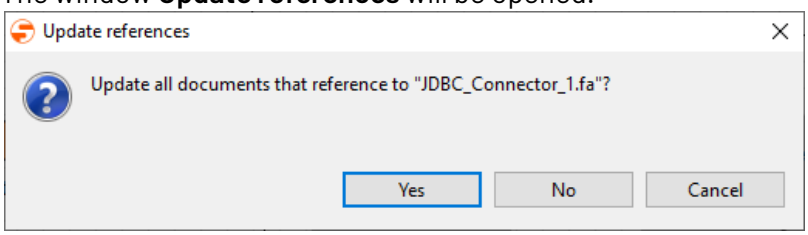
4.4.4 Moving elements and updating references automatically

You can move any element within the current project of the Repository. References to the moved file in other processes can be updated automatically.

1. Move the resource to the target in the repository via drag&drop.

   > 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

   The window **Update references** will be opened.



2. If you want to update references to the moved file or folder automatically:

- Click **Yes**
  The window **Confirm Move** will be opened. It displays which repository elements use the moved file.
- Click **OK** to apply changes.
  ? Unknown Attachment When moving XSL mappings that have referenced input or output XML documents/schemas, those file references must be modified if their relative paths have been changed.

3. If references should not be updated: Click **No**.

> If you press **Ctrl/Shift**, you can select multiple files in the **Repository Navigator**. For each file, the **Update references** dialog will be opened. It offers additional options:
>
> - **Yes To All:** Update references for all files to be moved
> - **No To All:** Do not update references

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

4.4.5 Repairing Repository References

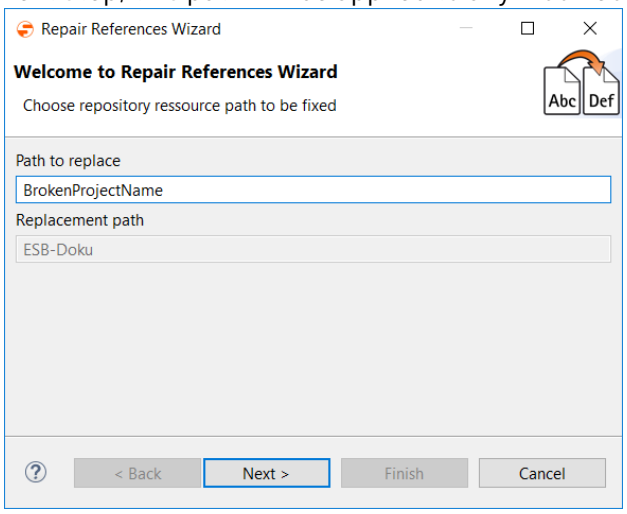
You can repair references to process components using the **Repair References Wizard**. This could be necessary if process diagrams have incorrect repository references – for instance because a project or folder has been renamed or it has been created from a legacy template.

1. Open a Technical Process, Business Process or Process Landscape where process component references are broken (greyed out), and retrieve the incorrect path of a selected process component from the **Properties** view.

   > **Example**
   > In a diagram, process components might be referenced incorrectly: its repository paths refer to a non-existing project `BrokenProjectName`. These references need to be corrected to the actual project name `ProjectName`.

2. Right-click on the project or folder in the Repository Navigator to open the context menu.
3. Select **Repair References** to open the **Repair References Wizard**.
   In **Replacement path**, the selected project or folder (i.e. the target name) is displayed. In the

next step, this path will be applied to any incorrect references.



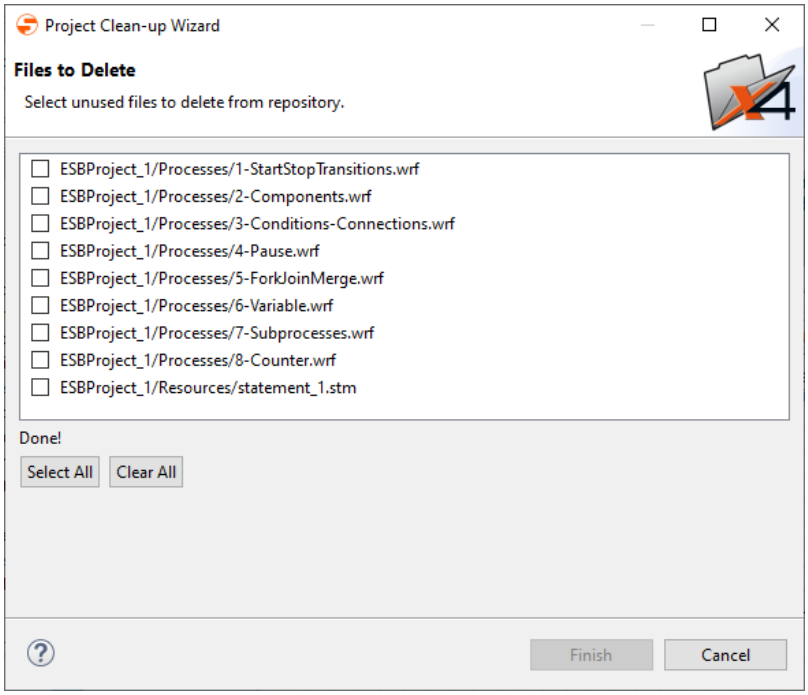
4. In **Path to replace**, enter the incorrect project or folder name that you have retrieved in step.
5. Click **Next** to display a list of processes that contain incorrect references and should be modified.
6. Click **Finish** to apply the changes.

4.4.6 Deleting Unused Resources

If the Repository contains unused resources, you can use the **Project Clean-Up Wizard** to list and remove them at the push of a button.

1. Right-click on the desired project in the Repository Navigator to open the context menu.
2. Select **Clean-Up Project** to open the **Project Clean-Up Wizard**.
   All non-referenced repository files are now listed.

> This process may take some time.



3. Select the desired resources.

> **Attention: Possible data loss!**
> Be aware that also files can be listed that are for example used as dynamic parameters in an XSL mapping.
> Only delete files when you are sure not to need them anymore!

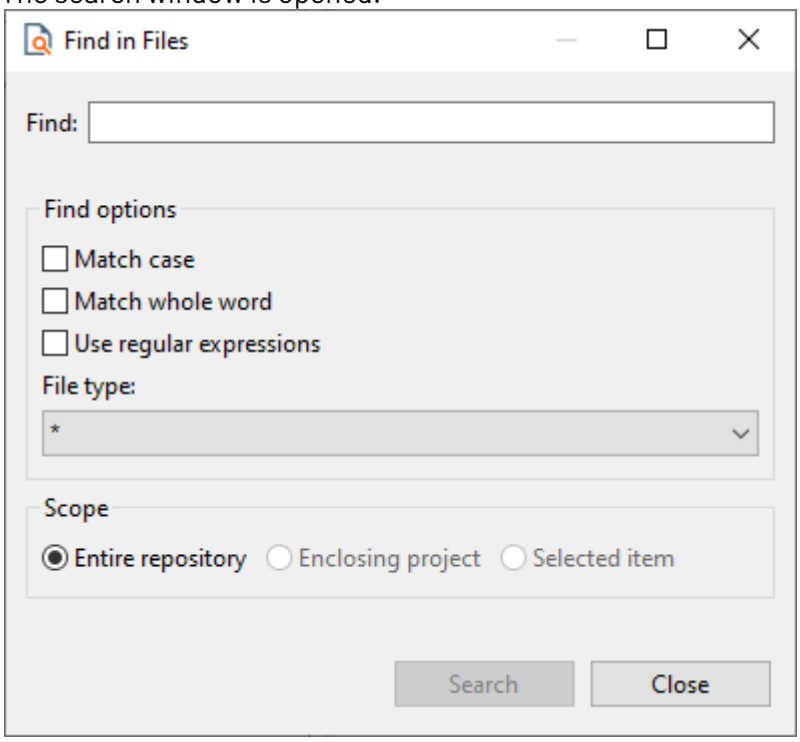
4. Click **Finish** and confirm deleting with **Yes**.
   The resources are now deleted.

4.4.7 Searching Repository Elements

Learn how to search for text, regular expressions and UTF-8 symbols within the repository using the **Find in Files** function.

> Note that only file contents are searched. File names are not included in the search.

1. Select menu **Edit** > **Find in Files**.
   The search window is opened.



2. Enter the characters to search for in **Find**.
3. Specify search options:
   - **Match Case:** The search is case-sensitive.
   - **Match whole word:** Search for the entire word.
   - **Use regular expression:** Search for regular expressions.
   - **File type:** Specify the file type to be searched, e.g. `bpm` or `fa`.

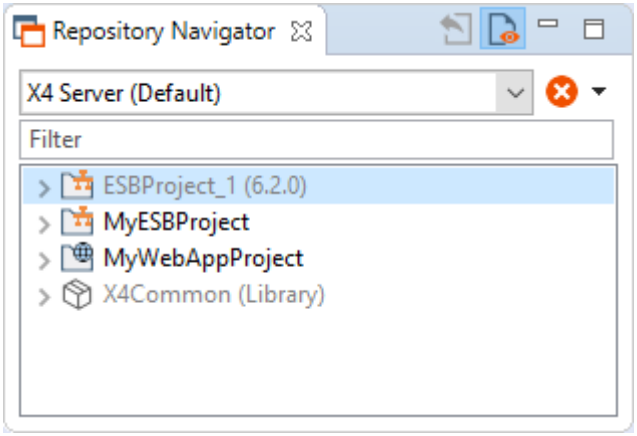
     With the setting * all file types are searched.
4. Define the search scope:
   - **Entire repository**: Search within the entire repository.
   - **Enclosing project:** Perform the search only within the selected project.
   - **Selected item:** Limit the search to the selected elements.
5. Click **Search**.

You can also search by adapter parameters, for example, to display all adapters with a specific parameter.

4.4.8 Migrating Outdated Projects

Outdated projects are specially marked within the Repository Navigator and are only displayed in a read-only view. To further process the projects, the X4 Designer offers a function for migrating the project to the current X4 BPMS version.

To do this, select the desired project in the Repository Navigator and migrate the project via the context menu **Migrate Project** or the menu **Project > Migrate Project**.



Outdated projects can be migrated to the current X4 BPMS version beginning with version 6.0.0 of X4 BPMS.
Projects with a version older than 6.0.0 must first be manually updated to version 6.0.0.

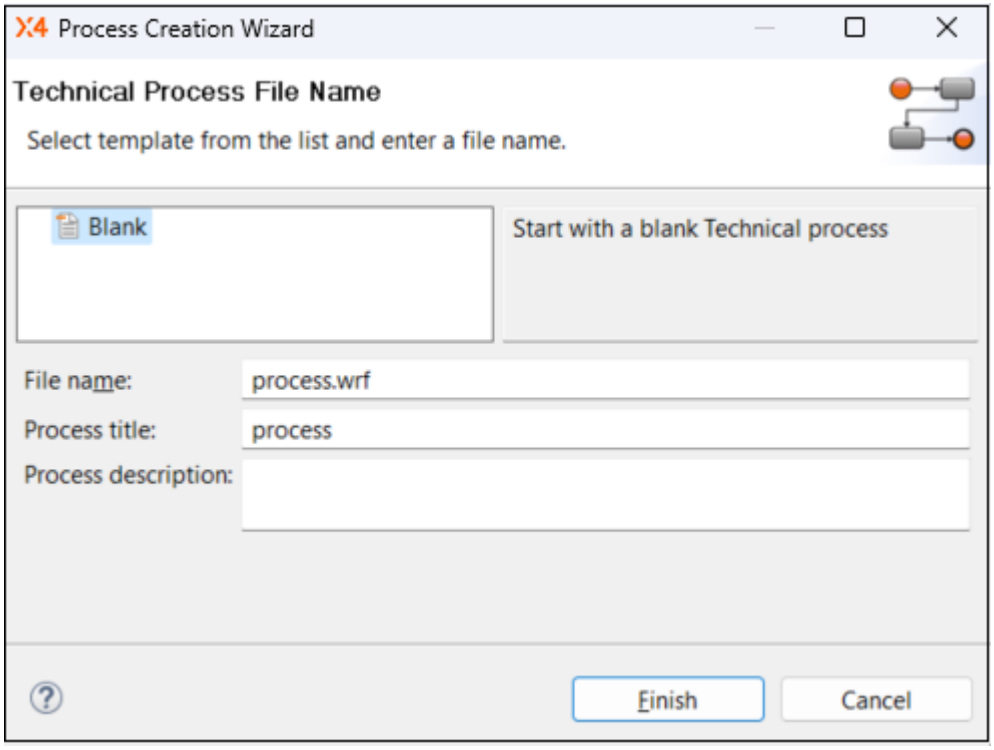
5 Creating and Modelling Processes

5.1 Creating Processes

The Process Editor enables you to model Technical Processes graphically.
You can drag both empty components from the palette and files from the **Projects** view to the drawing area and configure their properties in the **Properties** view in order to create executable processes.

To create a new process, proceed as follows:

1. Right-click on the **Processes** folder and choose **New > Technical Process...** from the context menu.
   The **Process Creation Wizard** dialog opens.



2. Select an empty or an already available template.
3. Enter a file name in **File name**.
   Spaces and special characters are not allowed in the file name!
4. Enter a process title in **Process title**.
   The process title is displayed in the process diagram and can be changed in the **Properties** view via the `Title` property. Moreover, you can use the `Show Title` property to show or hide the title.
   To display the `Show Title` property, choose the **Show Advanced Properties** icon in the **Properties** view.
5. Enter a description in **Process description**. The description is optional.

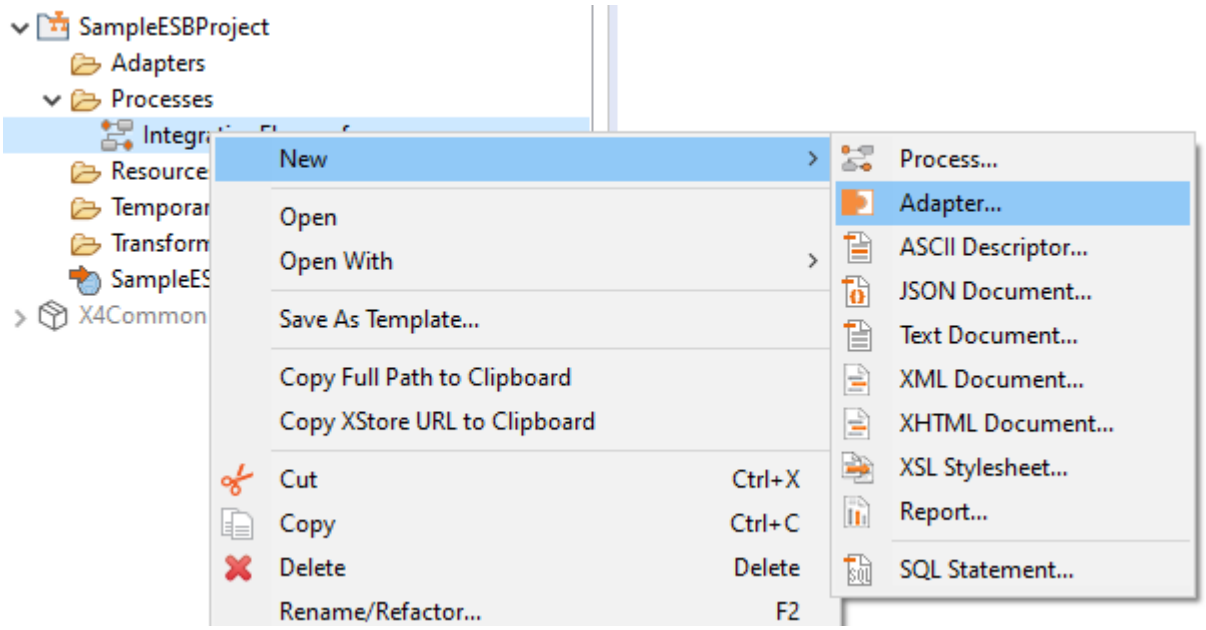
6. Click **Finish** to create the process.
   An empty process diagram with the file ending `.wrf` is created in the **Projects** view and opened within the Process Editor.
7. Model the Business Process as desired:
   - Drag components from the **Palette** or files from the **Projects** to an existing transition between two process components or to a free space in the drawing area.
   - If needed, draw transition lines to connect process components. To do so:
     - Move the mouse cursor over the source component to display a small arrow.
     - Click on the arrow, and drag a line to the target component to create a sequence flow.

> **Tipp**
> An integrated process validator allows the process to be checked for validity via the **Project > Validate Project** menu.
> If you use files from the Projects view in your process and configure all properties in the **Properties** view correspondingly, you can execute the process by clicking ▶.

5.1.1 Using Private Files Below Processes

If process resources are to be used only within a particular process, they can be created directly below a parent process. This means that they can only be used in the process context and cannot be called from other processes.

Basically, the same resources can be created below an individual process as within a project – be it sub-processes, documents such as XML files, SQL statements or even adapters.



The resources to be used can be created both, via the context menu **New > <*Resource name*>** and via the toolbar **> Add <*Resource name*>**.
The resources created can then be parameterizes as usually and integrated into the process diagram via drag & drop from the repository.

5.2 Properties of a Technical Process (.wrf)

Technical Processes (`.wrf`) are modelled graphically using the Process Editor. Processes can be executed on the X4 Server, if all components were parameterized and linked correctly with repository files.

You can edit the properties of a process in the **Properties** view, by clicking on a free area of the Process Editor's workspace.
The most common properties are displayed by default, however, additional settings can be made using the Properties view toolbar:

- *Show Categories*: Show properties sorted by categories
- *Show Advanced Properties*: Show advanced properties

Properties

Documentation	
`Title`	Process title; Can be changed, if required
`Show Title`	Show or hide process title by enabling or disabling the checkbox
`Description`	Process description; Can be changed, if required
`Version`	Project version If no version is set, the time stamp is taken as version.
`Project Name`	Project name; Can be changed, if required
`Created`	Creation date of the process
`Created by`	Creator of the process
`Last Update`	Date of the last change
`Last Update by`	Name of the last editor
Metadata	
`Manage Metadata`	Manage metadata, e.g. add, delete or edit; To do this, select the property and click , see Prozess-Metadaten hinzufügen
Technical	
`Schema Declaration`	Define schema for input and output of the process

`Location`	Complete relative path of the process in the X4 repository; Is set automatically
`Can Stop`	Process can be stopped at runtime This property is not inherited by subprocesses. **Possible values:** • *`true`:* For typical processes (default) • *`false`:* For critical processes that need to be completed
`Stop on error`	Stop the process execution in case of an error **Possible values:** • *`true`:* Stop process execution, if an error occurs (default) • *`false`:* Continue erroneous process (if possible)
`In service`	Set process active or inactive. If inactive processes are executed, they receive the process ID `-1`. Subprocesses (processes called from other processes) can always be executed – independently from this property. **Possible values:** • *`true`:* Process can be executed (default) • *`false`:* Process cannot be executed (only called as subprocess)
`Schedule`	Time scheduling tool enabling the scheduled execution of processes on the X4 Server without X4 Designer; To schedule a process, select this property, and click .... **Possible values:** • *`Disabled`*: No schedule was defined (default). • *`Enabled`*: A schedule for this process was defined.

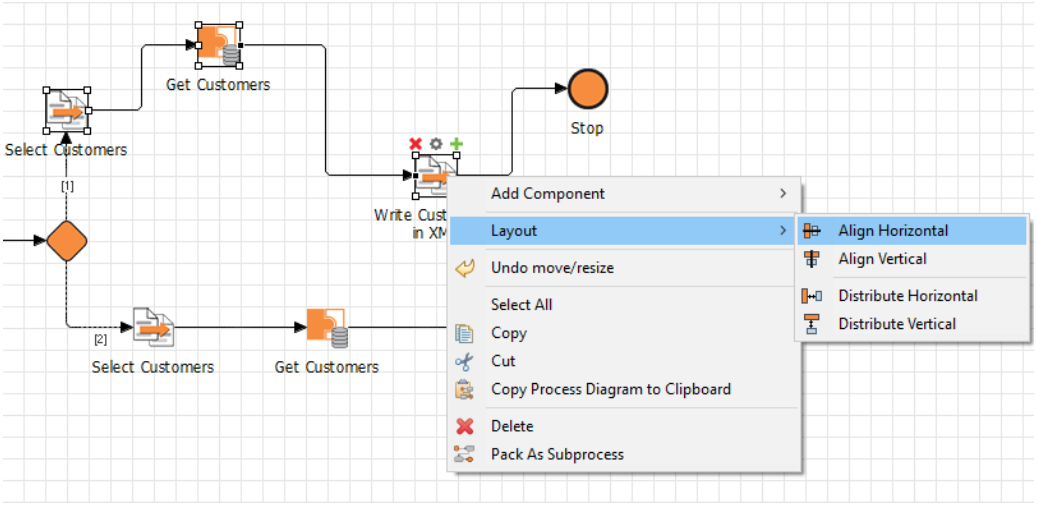
`Instance limit`	Maximum number of instances of this process that can be executed at the same time **Please Note!** `Instance limit` is ignored, if the process is executed as subprocess. **Possible values:** • Any positive integer number • `0`: No limitation (default)
`Owner`	Free text area to name the business process owner

5.3 Aligning Process Components

Process components of Technical Processes and Business Processes can be aligned easily via the context menu entry **Layout**. This allows you to generate a clean and structured model in quick and easy way.

To align process components, select the required components and open the context menu by right-clicking. Then select **Layout** and the desired option. The following options are available:

- `Align Horizontal`: Align the process components horizontally
- `Align Vertical`: Align the process components vertically
- `Distribute Horizontal`: Distribute the process components horizontally
- `Distribute Vertical`: Distribute the process components vertically



For the options `Align Horizontal` and `Align Vertical` at least two process components must be selected, for the options `Distribute Horizontal` and `Distribute Vertical` at least three.

5.4 Validating Processes

Both Business Processes (`.bpm`) and Technical Processes (`.wrf`) can be checked for validity via an integrated process validator. The validation can be performed based on default rules as well as based on custom validation rules.

The process validation can be initiated via the following menus:

- Context menu **Validate** when right-clicking on processes within the X4 Repository
- Menu **Project** > **Validate Project**

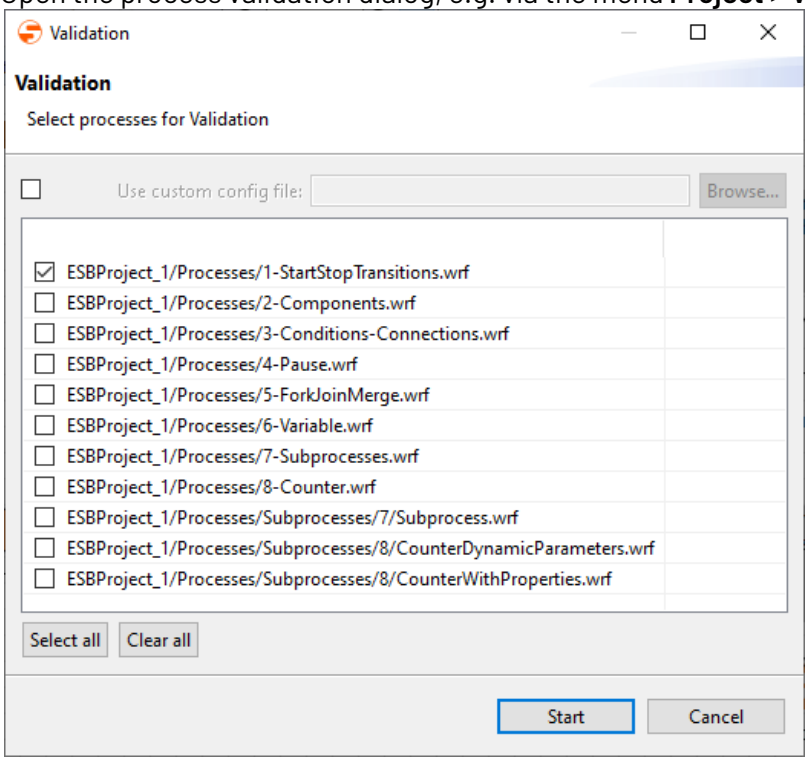
There are three severity levels for the process validation:

- `Error`: Errors preventing the process execution
- `Warning`: Errors that may cause the process to run incorrectly
- `Information`: Information regarding the process' state

5.4.1 Checking Processes

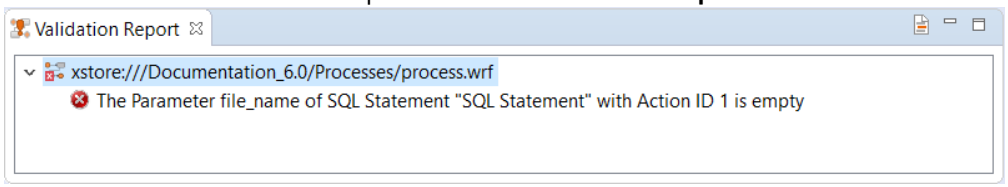
To check processes for validity proceed as follows:

1. Open the process validation dialog, e.g. via the menu **Project** > **Validate Project**.



2. Select the processes to be validated.
3. If required upload a configuration file for custom validation rules via **Use custom config file** and **Browse**, see section *Custom validation rules*.

4. Click **Start** to begin the validation.
   The validation result will be opened in the **Validation Report** view.



- By double-clicking on the process path the corresponding process will be opened.
- By double-clicking on a validation result the properties of the corresponding process component will be opened in the *Properties* view.

5.4.2 Default Validation Rules

The following validation rules are already defined by default:

Validation rule	Description	Severity
`Action Parameter Validation`	Are all actions parameterized correctly?	Error
`Condition Validation`	Are all branches of a `Condition` component filled?	Warning
`Connection Validation`	Are all paths closed?	Error

5.4.3 Custom Validation Rules

It is also possible to define custom validation rules. They must be provided as XSLT mapping with a pre-defined structure.

Mapping example for custom validation rules

```
<?xml version="1.0" encoding="UTF-8"?>
<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>
```

An XML configuration file with the following structure must be uploaded for the process validation. The mapping to be used is referenced here by means of its `xstore` URL:

```
<?xml version="1.0" encoding="UTF-8"?>
<validationConfig>
    <default active="true"/>
    <custom_xslt active="true" name="processMetrics">
        xstore://Projekt/Unterverzeichnis/customValidationMetrics.xsl
    </custom_xslt>
</validationConfig>
```

5.5 Parameterizing Process Components

The following pages provide information on:

- the available types of parameters
- how to parameterize or configure the process components of Technical Processes
- how to set parameters in a Technical Process

Read more about this on the following subpages:

- Parameterization Options
- Use Cases for Parameters
- Setting Dynamic Parameters
- Using Component Parameters
- Setting custom states with ConditionState
- Using Parameters in XSL Mappings

5.5.1 Parameterization Options

Here you get an overview of all options to parameterize a process component.

Scope of parameters

- **System parameters**: Parameters provided by the system (see System Parameters)
- **Global parameters**: Parameters that you can create as a user and that can be used in all projects of the X4 BPMS (see Global Parameters)
- **Project parameters**: Parameters that can be created in a specific file in a project, used within this project and imported or exported with the project (see Project Parameters)

Using parameters

Parameters can be used in Technical Processes in process components, mappings, adapters, and conditions. In business processes (BPM processes), parameters can be used in rules and human tasks. Global, system or project parameters can be used.

With the shortcut `Ctrl+Space` all available parameters (global, system and project parameters) are displayed in process components, mappings, adapters, and conditions.

Note the following prioritization when using parameters:
System parameters → Project parameters → Global parameters.
In case of name conflicts, an error message or warning is displayed in the **Problems** view:

- Error message in case of name conflicts of system and project parameters
- Warning in case of name conflicts of project parameters and global parameters

Where are parameters saved?

System parameters	The X4 BPMS contains a number of predefined system parameters, e.g. `%PID%` (current process ID) or `%NOW%` (current date and time). System parameters cannot be changed during runtime. **Typical use case:** • Set technical parameters, e.g. a mail server address, for this runtime environment/stage • Output a file name or the process ID of a technical process
For additional use cases not covered by system parameters, you can create global or project parameters. Global or project parameters, unlike system parameters, can be set at runtime.	
Global parameters	You can define global parameters in the **Global Parameter Configuration**. This configuration is stored in the respective process. For examples of use cases of global parameters, see Using global parameters. **Typical use case:** Set parameters that are used in different projects and that can be changed during runtime, e.g. the recipients of system-generated mails or the footer in system-generated mails.

Project parameters	Project parameters can be used in multiple Technical Processes in a project. These parameters are stored in the **Project Parameter Configuration** (`*.param`) in the project and can be exported or imported with the project. **Typical use cases:** • Technical Process: A project parameter can be defined as a limit for measured values (measured value must be between two limits). • Business Process with Business Rule: A project parameter can be defined as a limit for checking a value (acceptance/ rejection depending on the set value). • Business Process with Human Task: A project parameter can be set in a mapping in a Technical Process that is called by the business process. Here, for example, the duration in the Human Task can be different depending on the environment.

Where are parameters used?

In Technical Processes, parameters can be set for the following components:

- Adapters
- Conditions
- Mappings
- further process components

In business processes (BPM processes), parameters can be set for the following components:

- Rules
- Human Task (via mapping in a Technical Process)

When parameterizing, the following priority cascade is applied:

- **Component Parameters** overwrite the process component configuration from the repository (e.g. an adapter) during process runtime.
- Dynamic parameters overwrite set parameters (global, project and system parameters) as well as the process component configuration from the repository (e.g. an adapter) during process runtime.

Action	Description
Parameters in Adapters	Parameters can be used in the configuration of adapters. **Typical use case:** In an **XML Encryption (Password)** adapter, a path to an element that is to be encrypted/decrypted can be set in the XPath parameter. This path can be fixed (`//Data`) or set via a parameter.
Conditions	You can define case distinctions in branches of **condition** components with parameters (see Creating and using conditions). **Typical use case:** It is to be checked whether the output of a process contains a certain parameter value.
Mappings	Parameters (global, project and system parameters) can be used in XSL mappings. For being available for use, they have to be included in the mapping (see Using parameters in XSL mappings).
Component Parameter	You can adjust selected components of an open process in the **Properties** view. With **Component Parameters,** you can parameterize a marked component for the individual process (see Component Parameters). These parameters are stored in the respective process. **Typical use case:** **Component Parameters** can be used to overwrite the parameters of an adapter in a process. As **Component Parameters** you can use fixed values (fixed path e.g. `//Data`) or parameters. System parameters or previously defined global or project parameters can be used here.

Action	Description
Dynamic Parameter	You can use dynamic parameters to overwrite global, system, and project parameters in XSL mappings. Within a process, you can use dynamic parameters to control different process modules depending on the processed data. Dynamic parameters are stored in XSL mappings in the form of processing instructions and interpreted by the subsequent process module, e.g. an adapter (see Setting dynamic parameters). The configuration is only changed dynamically during the process runtime. **Typical use case:** You can use dynamic parameters to control process modules and overwrite parameters that have already been set at process runtime.

5.5.2 Use Cases for Parameters

Here you can find possible application examples for the use of global, system and project parameters.

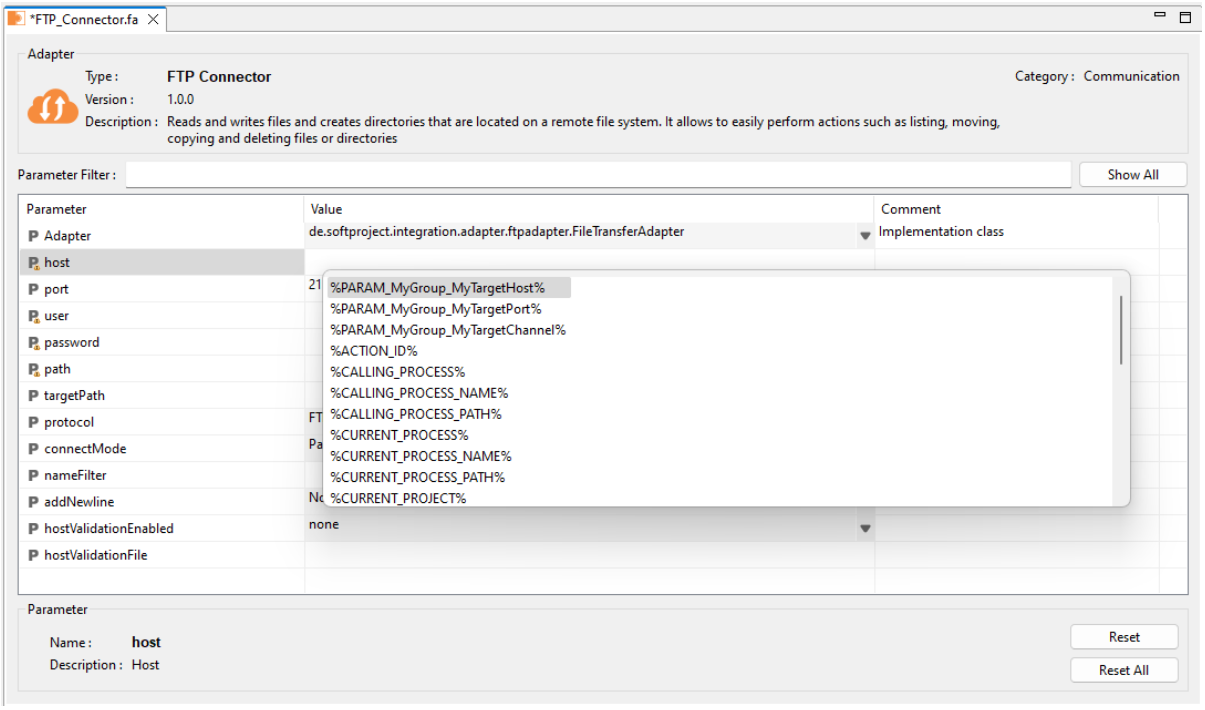
Using parameters in adapters

Parameters can be set in two ways in adapters:

- in the **Properties** view via **Component Parameters** (see Component Parameters)
- in all text fields within the Adapter editor

> **Note**
> Please note the following when using parameters in adapters:
> With the shortcut `Ctrl+Space` all available system parameters as well as global and project parameters of type **String** are displayed in the `Value` column.

The corresponding parameter can be selected from the displayed drop-down list:



5.5.3 Setting Dynamic Parameters

How can I set parameters dynamically?

In a technical process, you can set parameters dynamically to control process modules depending on the processed data and, if necessary, overwrite parameters already set at process runtime.

Dynamic parameters are generated using XSL mappings that output specific processing instructions. These processing instructions are processed by a subsequent process component (e.g.To do this an adapter).

To do this, you have several options:

- Set parameters via single XSL mapping: Here you can create a single XSL data mapping that transforms XML data and contains dynamic processing instruction definitions at the same time.
- Parameterize dynamically without data transformation: You can also create a separate XSL mapping that generates processing instructions only.

Note
Observe the following prioritization when using **Component Parameters** and setting parameters dynamically:

- **Component parameters** overwrite the configuration of process modules in the repository (for example, function adapters) at process runtime.
- Dynamic parameters overwrite set parameters (global, project and system parameters) as well as the configuration of process modules in the repository (for example, function adapters) at process runtime.

5.5.3.1 Setting Parameters Dynamically via XSL Mappings

1. Place an XSL mapping before the adapter or database connection to be parameterized in the Technical Process.
2. Define a processing instruction in the XSL mapping according to the following pattern: `<xsl:processing-instruction name="DynamicParameter">ParameterName=" ParameterValue "</xsl:processing-instruction>.`
3. Replace `ParameterName` with the required parameter name.
4. Replace `ParameterValue` with a value, in most cases using an XSL instruction.
5. If the XSL mapping shall create dynamic parameters only, define an additional processing instruction `<xsl:processing-instruction name="NoResult"/>`. As a result, the XSL mapping will not produce an output document.

Note
The XSL mapping must generate a root element in each case in order to be able to execute the mapping.

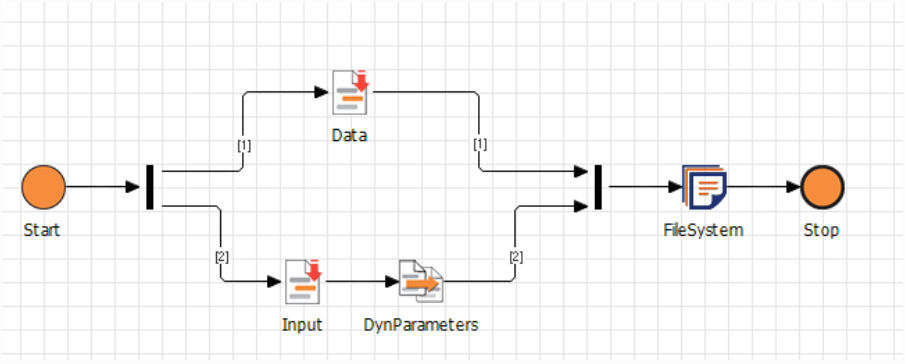
5.5.3.2 Parameterizing Dynamically without Data Transformation

If you want to hand over an input document without transformation to a dynamically parameterized adapter, insert a `Fork` component into a Technical Process diagram:

1. In the first branch, set the XSL mapping according to the above described procedure.
2. In a second branch, set the process component to be handed over.
3. Place the adapter right after the `Join` component.

Example

A file name stored inside an XML document is handed over to a File system transfer adapter as dynamic parameter.



In this example, the XSL mapping contains the following code:

```
<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 Dynamic Parameters with DynamicParameter

With XSL mappings, you can control the process execution at runtime. An XSL mapping implements the logic to customize the configuration of adapters, transfers, and database connections. Depending on the processed data, custom parameter values can be set. Those custom parameter values will be processed by the following adapter or database process component and overwrite parameter values set in the component.

Creating Parameter Values

To set parameter values within Technical Processes dynamically, create an XSL mapping that produces a specific Processing Instruction which will be handed over to the following process component and will be interpreted.

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

Output of the XSL mapping:

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

Parameters and parameter values

In the XSL mapping, you can define any number of different parameters and corresponding values:

- `parameterName`: Name of the parameter to be overwritten
- `parameterValue`: Parameter value to be overwritten; always inside of quotation marks

> **Note**
> Please note that the XSL mapping must generate a well-formed XML document providing a root element, in order to be able to execute the XSL mapping.

Example

```
<?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 Using Component Parameters

How can I use Component Parameters?

For certain technical process components (XSL mappings, PDF Creator components and adapters), you can define component parameters in a process.

As component parameters, you can use a set of parameters that are interpreted during process runtime.

These parameters can be set:

- system parameters, e. g. `%PID%` (current process ID) or `%NOW%` (current date and time). For a complete list of available system parameters, see Overview of system parameters)
- global parameters (see Global Parameters)
- project parameters (see Project Parameters)

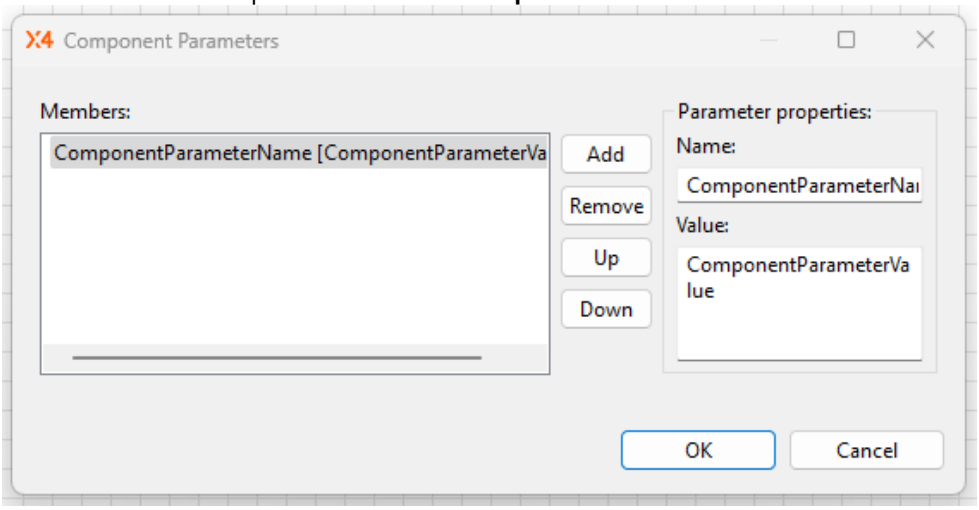
Note
Observe the following prioritization when using **component parameters** and dynamic parameters:

- **Component parameters** overwrite the configuration of process components in the repository (for example, adapters) at runtime.
- Dynamic parameters overwrite set parameters and the configuration of process components in the repository (for example, adapters) at process runtime.

5.5.4.1 Defining Component Parameters

Proceed as follows to define component parameters:

1. Open the window **Component Parameters**:
   - In the technical process, select the process component to which you want to add a component parameter.
   - In the **Properties** view, click **Component Parameters** to display the button.
   - Click to open the window **Component Parameters**.



2. In **Name**, enter the name for the component parameter .
   - In XSL mappings and reports you can retrieve these parameters as global XSL parameters, e. g. a global parameter `PDFPath` can be referenced by the XSL mapping with `<xsl:param name="PDFPath"/>`.
   - For adapters and database connections: Use the exact (technical) name of the parameter to be overwritten.
3. In **Value**, enter the parameter value to be overwritten during runtime.
   Here, you can insert parameters that will be replaced by current values during process runtime, e. g. `%PID%` is replaced by the current process ID.
   For a complete list of available system parameters, see System Parameters.
4. Click **OK** to save the component parameters in the process.

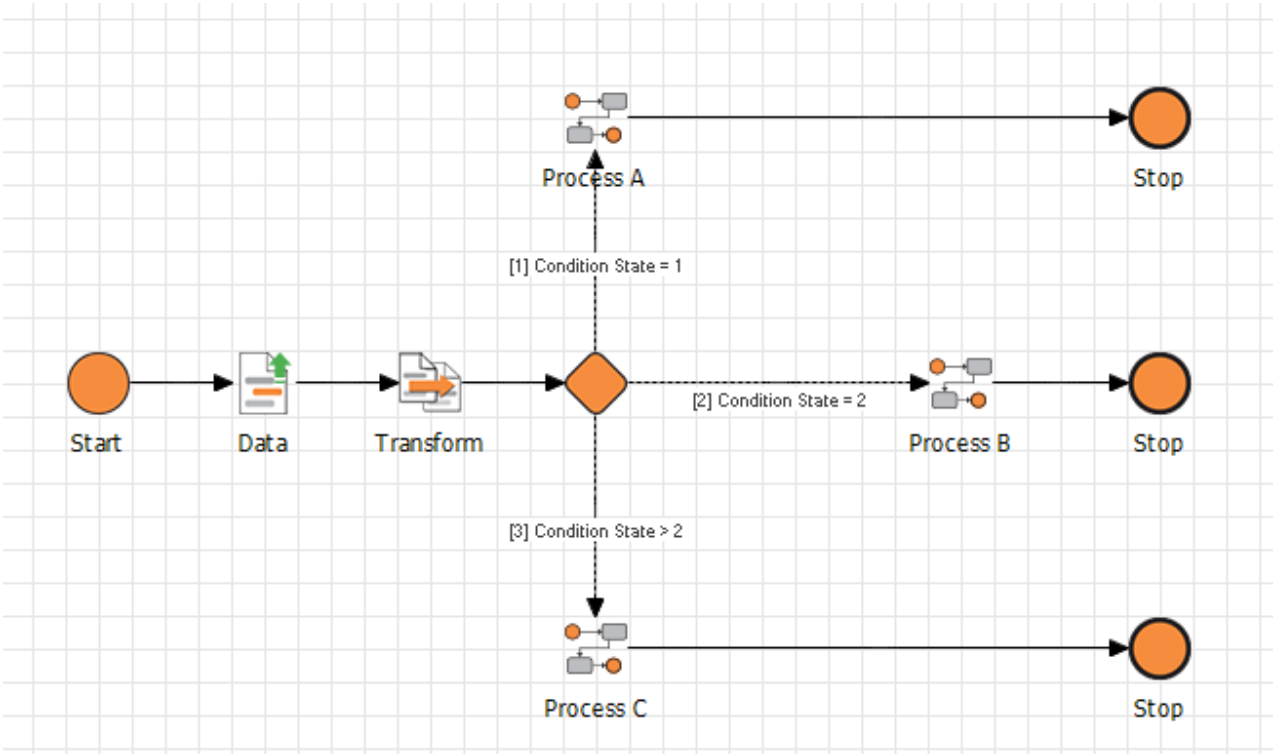
> **Note**
> If a component parameter is defined for a certain process component, the component is marked with a .

*Available functions in window **Component Parameters**:*

- **Add**: Create additional component parameters
- **Remove**: Delete a selected parameter from the list
- **Up / Down**: Change the parameter order in the list

5.5.5 Setting custom states with ConditionState

An XSL mapping can produce status information depending on the processed data, using a processing instruction. This processing instruction can be interpreted by a `Condition` component during runtime of the Technical Process. Depending on the status, a corresponding process branch can be executed.



1. Place an empty XSL mapping in the Technical Process before a `Condition` component.
2. Define processing instruction(s) in the XSL mapping according to the pattern:

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

3. Replace `value` with a value from the processed XML document.
   Example: `<xsl:value-of select="data"/>`

   > **Note**
   > Allowed status values are integer numbers, and negative numbers (e. g. `-1`) for an error status; see also Status Values for Processes and Process Steps.
   > The following status values are not allowed, because they are reserved by the system:
   > - *-999* (Error status of the engine)
   > - *-998* (Process not executed due to license reasons)
   > - *999* (Status ID for a process component not yet executed)
   > - *998* (is used by the savepoint mechanism and causes the immediate termination of the entire technica
   > - *997* (Causes the immediate termination of the entire Technical Process)
   >
   > **Recommendation:** Use numbers greater than `1000` for custom state values, in order to avoid possible overlaps with other status values.

4. Make sure that the XSL mapping creates a root element, in order to be executed.
   The XSL mapping now outputs processing instructions according to the pattern `<? ConditionState value ?>`.
5. In the subsequent `Condition` component, define a status for each branch (see Creating and using conditions).
6. If you create other *ConditionStates* in the further process progress, remove the preceding Processing Instructions from the respective input XML document e. g. using `<xsl:copy-of>` to avoid unexpected behavior during process execution.

> **Problem solving: The XSL mapping cannot be executed?**
> Possible reasons: The processed value might be one of the impermissible state values *-999*, *-998*, *999*, *998*, *997* or a string or the XSL mapping produces an XML document which contains no root element.
> - Only use allowed state values (see Status Values for Processes and Process Steps.).
> - Add for instance an empty element `<root/>` after the ConditionState definitions in the XSL mapping.

5.5.6 Using Parameters in XSL Mappings

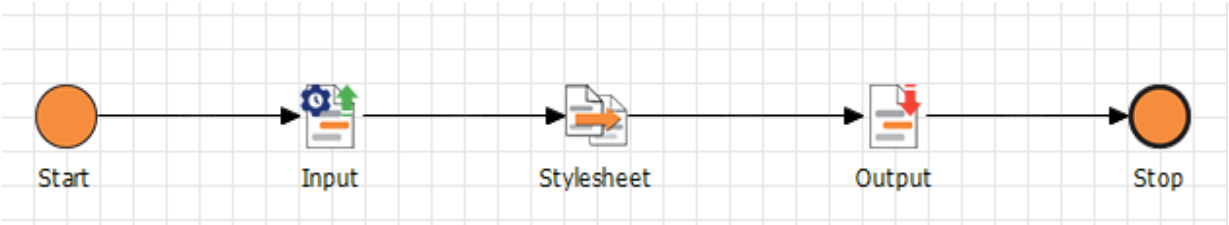
Using Parameters in XSL Mappings

You can use global, project and system parameters in XSL stylesheets.

Example for use

Using parameters in XSL mappings

The following example shows the use of parameters in a mapping. Before using them, the relevant parameters must be inserted in the XSL stylesheet.



Creating a mapping

The XSL mapping must be structured according to the following scheme. The parameters to be used must first be declared in the XSL stylesheet under their fully qualified name. Then they can be used as usual.

Example: 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>
```

Modelling the Technical Process

1. Create a new Technical Process via the menu **New > Technical Process**.
2. Drag a file for the input of the mapping into the process.
3. Drag the XSL mapping (see example) into the Technical Process.

4. Optional: Create a file to store the output of the mapping.
5. Save the Technical Process and then run it.

5.6 Creating and Using Parameters

When does the use of parameters make sense?

If technical processes are to be equipped for variable environmental conditions, parameters can significantly simplify process modelling.

What are possible use cases?

Parameters can be used to define any keys or value pairs: e.g. technical environment variables, such as paths, or technical parameters, such as limit values for checks. At runtime of the technical process, these parameters are replaced by the previously defined values. This allows technical processes to be modelled in such a way that they can be executed in different environmental conditions, for example for operation in staging systems. Technical processes can thus be controlled via external parameters without further adaptation of the process models.

Read more about the creation and use of parameters in the following chapters.

- System Parameters
  - Overview of System Parameters
  - Using System Parameters
- Global Parameters and Groups
  - Creating Global Parameters and Groups
  - Copying Global to Project Parameters
  - Using Global Parameters
- Project Parameters
  - Creating Project Parameters
  - Renaming Project Parameters
  - Creating custom types
  - Reusing Global and System Parameters

5.6.1 System Parameters

How can I use system parameters?

The X4 BPMS provides a number of system parameters that you can use in process modules. During the execution of the Technical Process, the parameters are replaced by the respective values and processed by the process engine.

Additional information:

- Overview of System Parameters
- Using System Parameters

5.6.1.1 Overview of System Parameters

The following parameters are provided with the X4 BPMS:

`%ACTION_ID%`	Number of the currently executed process action. *Example: `3`*
`%CALLING_PROCESS%`	Relative repository path and file name of the Technical Process that called the current Technical Process. This parameter can only be substituted, if the current Technical Process has been invoked directly from another Technical Process as subprocess. *Example: `Project/Processes/ProcessCaller.wrf`*
`%CALLING_PROCESS_NAME%`	File name (without file extension `*.wrf`) of the Technical Process that called the current Technical Process directly as subprocess. This parameter can only be substituted, if the current Technical Process has been invoked directly from another Technical Process as subprocess. *Example: `ProcessCaller`*
`%CALLING_PROCESS_PATH%`	Relative repository path of the Technical Process that called the current Technical Process directly as subprocess. *Example: `Project/Processes/`*
`%CURRENT_PROCESS%`	Relative repository path and file name of the current Technical Process. *Example: `Project/Processes/Process.wrf`*
`%CURRENT_PROCESS_NAME%`	File name (without file extension `*.wrf`) of the current Technical Process (`*.wrf`). *Example: `Process`*

%CURRENT_PROCESS_PATH%	Relative repository path of the current Technical Process. This parameter can only be substituted, if the current Technical Process has been invoked directly from another Technical Process as subprocess. *Example:* `Project/Processes/`
%CURRENT_PROJECT%	Name of the current respository project. *Example:* `Project`
%HOSTNAME%	Host name of the current *X4 Server*. *Example:* `YC-WS-000.yourcompany.local`
%HTTP_PORT%	HTTP port that has been defined in the configuration file `x4config.xml` within the element `<webContainerURL/>`. *Example:* `8080`
%INSTANCE%	Name of the current *X4 Server* instance that has been defined in the configuration file `x4config.xml` within the element `<instanceName>`. *Example:* `X4`
%IPADRESS%	IP address of the *X4 Server*. *Example:* `192.168.0.1`
%NOW%	Current date and time in the ISO format `yyyy-MM-ddThh:mm:ss`. *Example:* `2021-08-11T17:46:50`
%NOW_PLAIN%	Current date and time in the format `yyyyMMddhhmmss` aus. *Example:* `20210811174650`
%PID%	ID of the current Technical Process instance as an integer number that is unique within an *X4 Server* instance. Subprocesses are executed with the same Technical Process ID, however, not processes that have been started using the *Process Starter*. *Example:* `1628696804892237809` If multiple instances of the *X4 Server* are executed within one application server simultaneously, it is recommended to use the unique parameter `%UID%` instead of `%PID%`.
%PUID%	Outputs the unique UUID of the start process.

%START_PROCESS%	Relative repository path and file name of the Technical Process that has been started originally. If the current Technical Process has been started as a subprocess of another Technical Process, the path and file name of this main Technical Process will be displayed. *Example:* `Project/Processes/ProcessCaller.wrf`
%START_PROCESS_NAME%	File name (without file extension `*.wrf`) of the originally started Technical Process. If the current Technical Process has been started as a subprocess of another Technical Process, the file name of this main Technical Process that has created the current Technical Process context will be displayed. ⓘ When calling a Technical Process with the *Process Starter*, a new Technical Process context will be created. *Example:* `ProcessCaller`
%START_PROCESS_PATH%	Relative repository path of the Technical Process that has been started originally. If the current Technical Process has been started as a subprocess of another Technical Process, the path of this main Technical Process will be used. *Example:* `Project/Processes/`
%TIME_MS%	Number of milliseconds since January 1st 1970, 0:00 h. *Example:* `1628696810712`
%UID%	Name of the *X4 Server* instance `%INSTANCE%` and the current Technical Process ID (`%PID%`). Thus, the process instance can be identified explicitly, if multiple *X4 Server* instances are executed within one application server simultaneously. *Example:* `X41628696804892237809`
%USER_COLLECTION%	Repository folder path (absolute or relative) of the current user in the `X4DB`; the so-called *"user collection"*. This value is taken from the global configuration file `X4config.xml` from with the element `<repositoryPath>`. Here also a relative path can be defined. *Example:* `C:\X4\X4DB\1` or `./X4DB/1`
%USER_COLLECTION_ABSOLUTE%	Absolute repository folder path of the current user in the `X4DB`; the so-called *"user collection"* *Example:* `C:\X4\Server\X4DB\1`

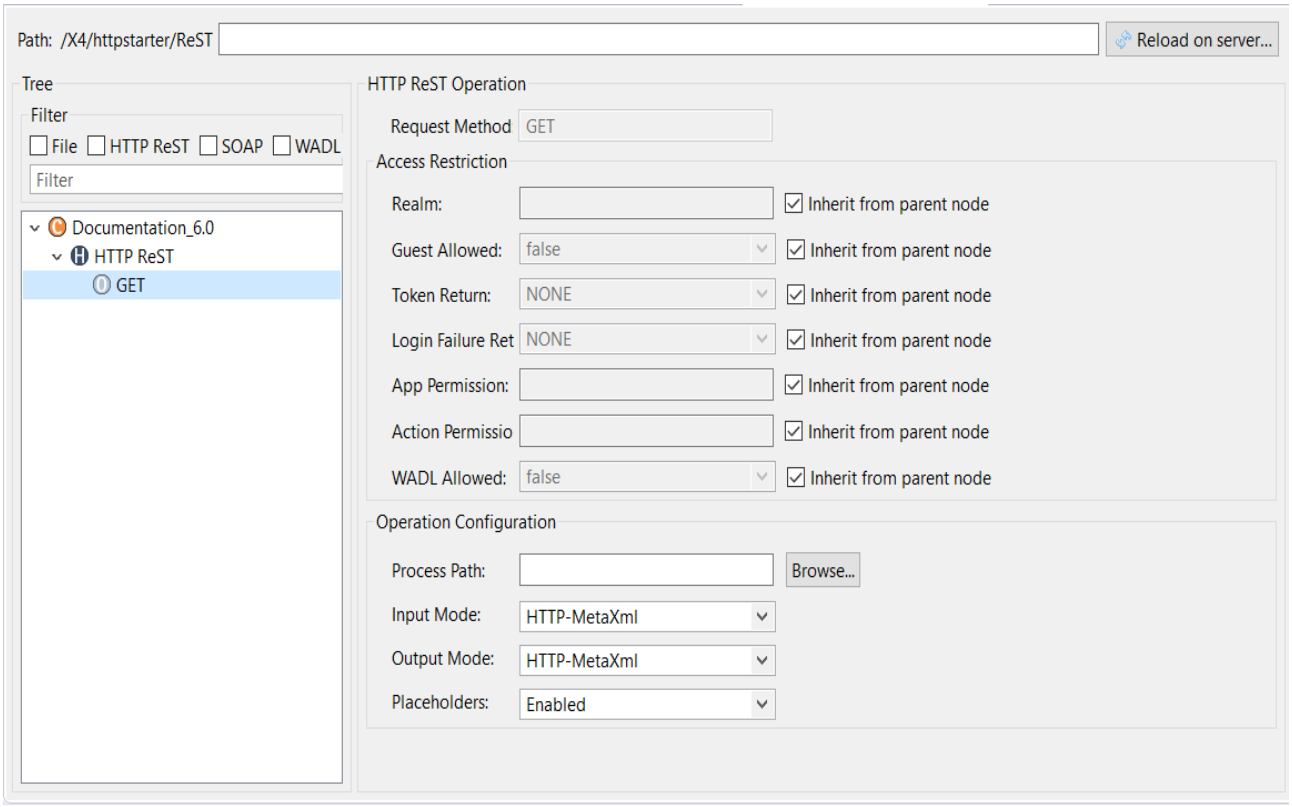
`%USER_ID%`	ID of the current user, whose repository folder contains the Technical Process ("*user collection*"). Since X4 BPMS version 4.0, the user ID is a string and no longer an integer number. *Example:* `1`
`%WORKING_DIR_ABSOLUTE%`	Absolute path of the current working directory, where the application server has been started. *Example:* `C:\X4\Server`

Parameter for Web Services

Within the Webservice Configuration Editor additional parameters can be enabled to be used exclusively for web services. This allows to enable parameter handling for this ReSTful webservice or SOAP webservice operation, and hand over HTTP request information to the Technical Process context as parameters (as a more efficient alternative to Input Mode `HTTP-MetaXml`).

Possible values:

- `Disabled`: Do not provide parameters within the Technical Process from the HTTP request (Default)
- `Enabled`: Provide parameters within the Technical Process context that contain HTTP request information



If option `Enabled` is selected, the following parameters with HTTP request information are available within the Technical Process context, e.g. within XSL mappings:

- `REQUEST_PARAMETER_<Name>`: Desired parameter (case-sensitive!)
- `REQUEST_HEADER_<NAME>`: Desired HTTP header (only uppercase!)
- `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`

These parameters can be used like default parameters, but cannot be overwritten.

5.6.1.2 Using System Parameters

System parameters can be used as follows:

Using system parameters in adapters

As with global and project parameters, system parameters can also be used in adapter configurations (see Use Cases for Parameters).

Using system parameters in XSL mappings

System parameters can be used as variables with `$Parameter` in XSL mappings, for example `$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 Global Parameters and Groups

How can I use global parameters?

Global parameters can be defined manually via the **Global Parameter Configuration**. Any parameter groups with global parameters can be created and managed directly in X4 Designer. These can then be used like system parameters and project parameters in the X4 Designer (Adapter Editor, Mapping Editor, Process Editor).

Additional information

- Creating Global Parameters and Groups
- Copying Global to Project Parameters
- Using Global Parameters

5.6.2.1 Persisting Global Parameters in Database

The plugin **SQL Parameter Storage** is available to persist global parameters in the database (see SQL Parameter Storage Plugin).

5.6.2.2 Managing Global Parameters via the Adapter Global Parameter Management

The adapter Global Parameter Management can be used to create and manage global parameters. The adapter offers various operations for managing global parameters, such as adding or deleting, but also importing and exporting parameters or parameter groups.

5.6.2.3 Controlling Global Parameters in the X4 Control Center

During operation, administrators can view and adjust the global parameter configuration in the X4 Control Center at any time via Administration > X4 API (Swagger UI). New process instances are then executed with modified parameters.

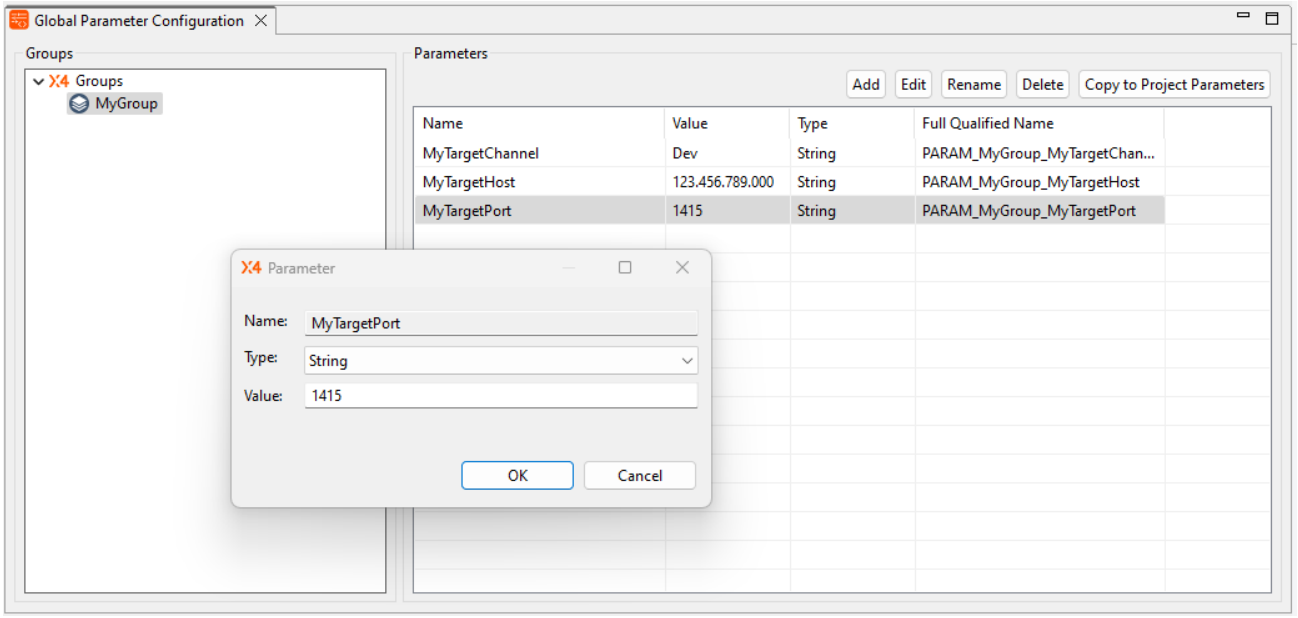
5.6.2.4 Creating Global Parameters and Groups

In the **Global Parameter Configuration**, you can create and manage global parameters and global parameter groups.

> **Note**
> Note the following prioritization when using parameters:
> System parameters → Project parameters → Global parameters.
> In case of name conflicts, an error message or warning is displayed in the **Problems** view:
>
> - Error message in case of name conflicts of system and project parameters
> - Warning in case of name conflicts of project parameters and global parameters

How to access the Global Parameter Configuration

Select the menu **Tools > Global Parameter Configuration**:



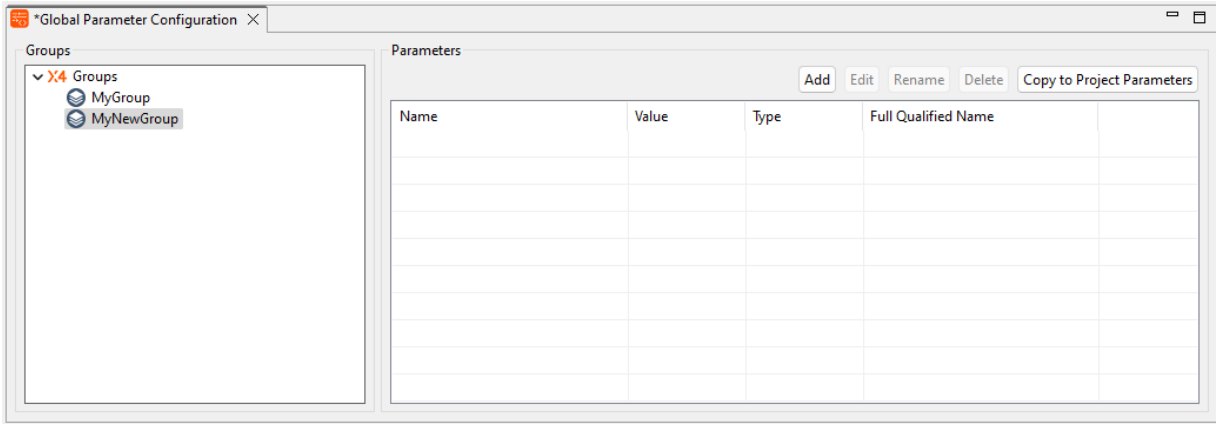
Creating a New Group with Global Parameters

> **Note**
> To avoid name conflicts, the name of the parameter group must be unique.

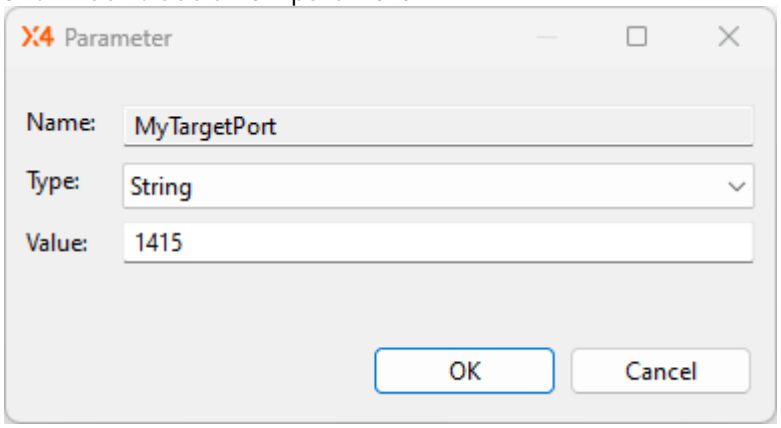
How to create a new group with global parameters

1. Open the **Global Parameter Configuration** via the menu **Tools > Global Parameter Configuration**.

2. Right-click on **Groups** to open the context menu.
3. Select **Add** in the context menu.
4. Enter a name for the parameter group and confirm with **OK**.
   The new group has been added and appears as a new node below the root node.
5. To add new parameters, select the group.



6. Click **Add** to add a new parameter.



7. Enter the properties for the parameter:

Data field	Description
Name	Name of the parameter **Note** To avoid name conflicts, the name of the parameter group must be unique. **Note** The following characters are allowed here: `^[a-zA-Z][a-zA-Z0-9.\-]*$` A dot is to be avoided if case you want to copy the global parameter to the project parameters. **Example**: `My-Global-Parameter-123`
Type	Data type of the parameter Possible values: • String • Boolean • Integer • Decimal • Date • Password **Note** For parameters of the type `Password`, the option **Show Password** can be activated to display the password unencrypted.
Value	Value of the parameter

8. Click **OK** to add the parameter.
   The parameter has been created; the fully qualified name of the parameter is created automatically.

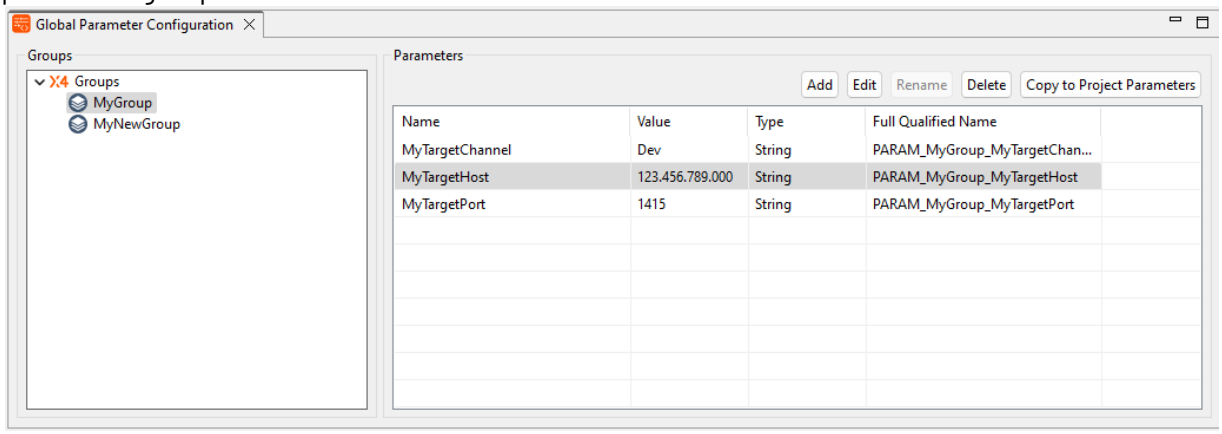
Note
Right-click on a global parameter to open its context menu. With **Copy Full Qualified Name to Clipboard** its fully qualified name can be copied and pasted elsewhere.

Renaming Global Parameters

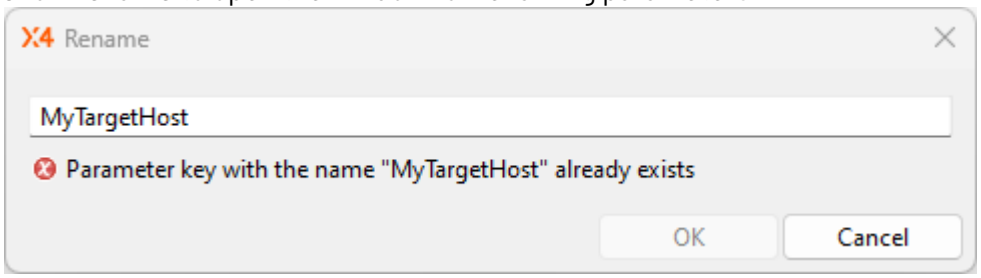
> **Note**
> The **Rename** action is active only when all changes are saved in the **Global Parameter Configuration**.

How to rename a parameter and automatically update its references in the process

1. In **Global Parameter Configuration**, select the desired parameter in the corresponding parameter group.



2. Click **Rename** to open the window for renaming parameters.



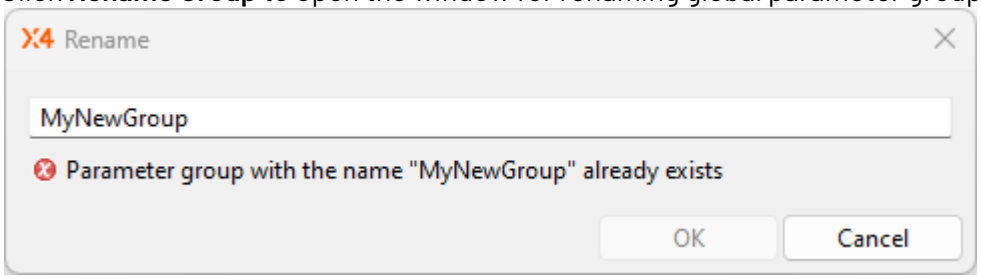
3. Enter a new parameter name.
4. Click **OK** to make the desired changes.

Renaming a Global Parameter Group

How to rename a global parameter group and automatically update its references in the process

1. In **Global Parameter Configuration**, right-click the desired parameter group to open the context menu.

2. Click **Rename Group** to open the window for renaming global parameter groups:



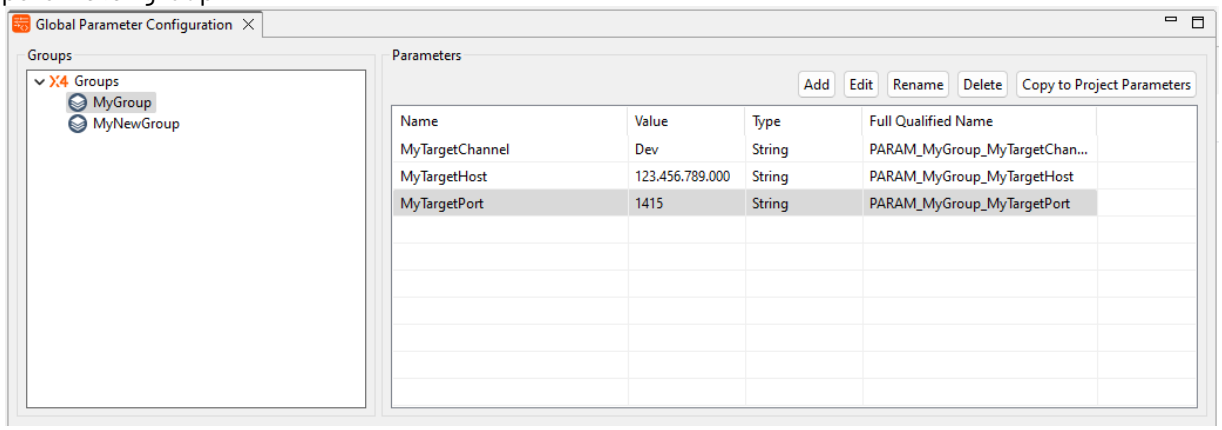
3. Enter a new name for the group.
4. Click **OK** to save the changes.

Editing Global Parameters

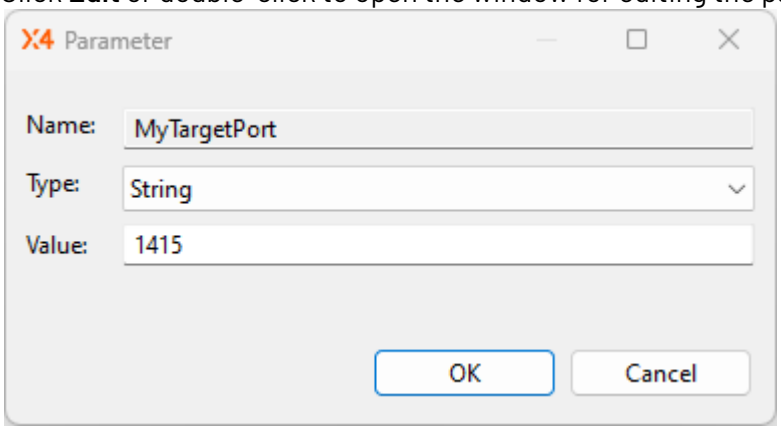
> **Note**
> By editing a global parameter, you can change the **Type** and **Value** of an existing global parameter.

How to edit a global parameter

1. In **Global Parameter Configuration**, select the desired parameter in the corresponding parameter group.



2. Click **Edit** or double-click to open the window for editing the parameter.



3. Edit the parameter.
4. Click **OK** to save the changes.

5.6.2.5 Copying Global to Project Parameters

Global parameters can be copied to project parameters.

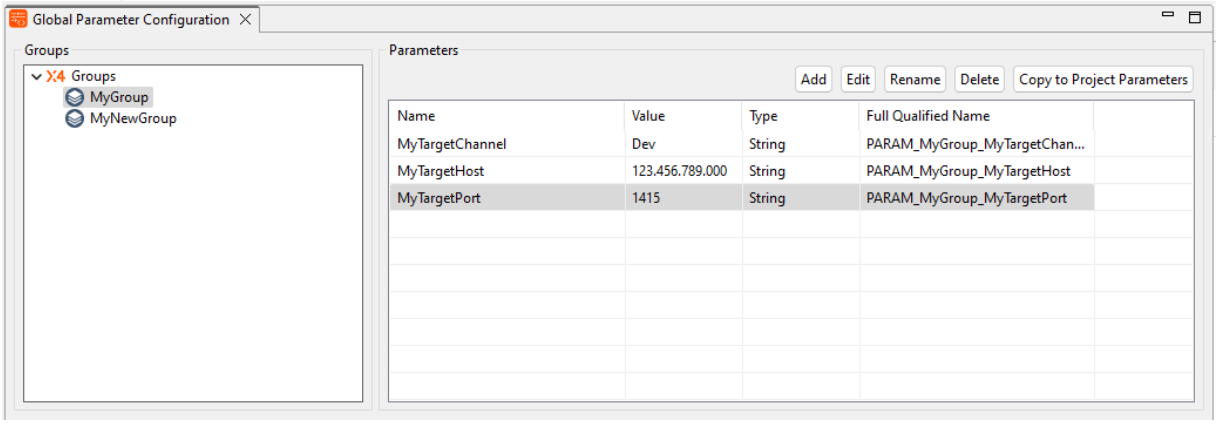
Global parameters are copied to root level, i.e. global parameters from global parameter groups are inserted in the **Project Parameter Configuration** at root level, i.e. without the group assignment from the **Global Parameter Configuration**.

> **Note**
> Observe the following prioritization when using parameters:
> System parameters → Project parameters → Global parameters.
> In case of name conflicts, an error message or warning is displayed in the **Problems** view:
> - Error message in case of name conflicts of system and project parameters
> - Warning in case of name conflicts of project parameters and global parameters

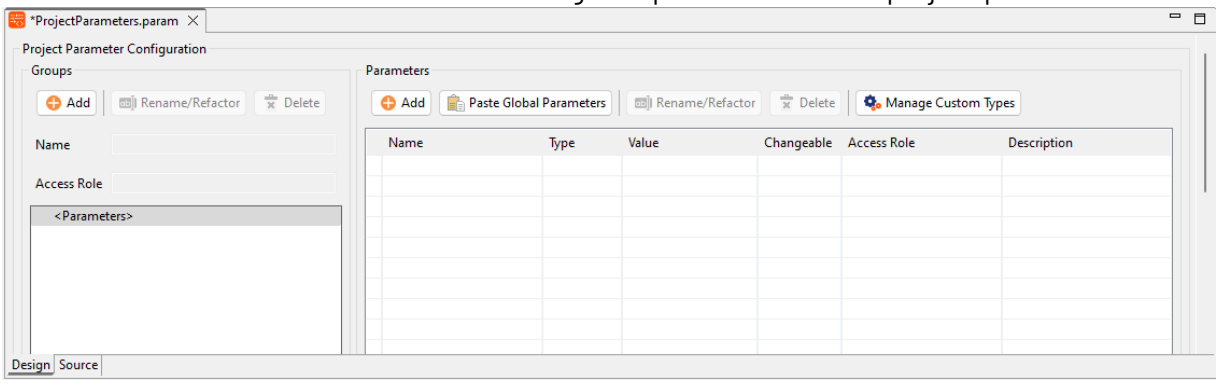
How to copy global project to project parameters

> **Note**
> We recommend deleting or rename parameters after the Copy to Project Parameters action: When a global parameter is copied to the project parameters, two parameters with the same name exist. A warning about this appears in the **Problems** view.
> Having parameters of the same name can however be confusing, for example, when selecting parameters in mappings, conditions, and adapters. Therefore, it is recommended to delete the global parameter or to rename the project parameter.

1. In **Global Parameter Configuration**, select one or more global parameters that you want to add to the project parameters.



2. Click `Copy to Project Parameters`.
3. In the project to which you want to add the global parameters, open the `*.param` file (**Project Parameter Configuration**).
4. Click `Paste Global Parameters` to add the global parameters to the project parameters.



The global parameters have now been added to the **Project Parameter Configuration**.

5.6.2.6 Using Global Parameters

How can I use global parameters?
Global parameters can be defined via the **Global Parameter Configuration**. Any parameter groups with global parameters can be created and managed directly in X4 Designer.
These can then be used just like system parameters and project parameters in the X4 Designer (in the editors of adapters, mappings, processes, and conditions).

Using Global Parameters in Adapters

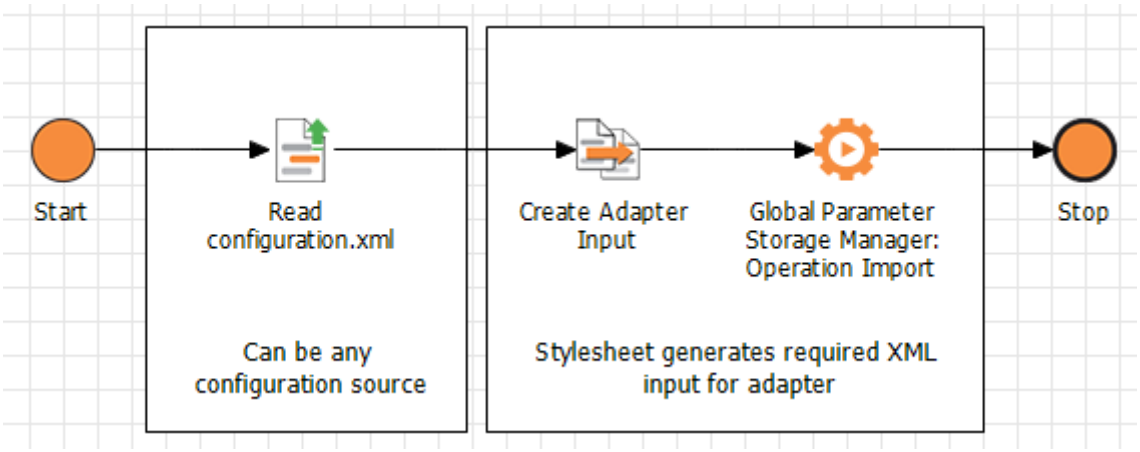
As with system and project parameters, global parameters can also be used in adapter configurations with `%PARAM_Group_Name%` (see Use Cases for Parameters).

Using Global Parameters in Technical Processes

For using global parameters in Technical Processes, the parameters to be used must first be read into the current process context. The Global Parameter Management adapter can be used to create global parameters via X4 processes and manage them at process runtime. The adapter offers various operations for managing global parameters, such as adding or deleting, but also importing and exporting parameters or parameter groups.

Creating global parameters with the Global Parameter Management Adapter

The following process example shows how global parameters can be created with a mapping via the Global Parameter Management adapter. The technical process reads in an XML document with configuration values, transforms it via a stylesheet and creates the parameters via the adapter with the appropriate operation. Thus, the parameters are available for the project and its processes.



Provide Input XML document

The input XML document must be structured according to the following scheme:

Example: Input XML with predefined parameters

```
<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>
```

Explanation:

- The root element `Placeholder` can contain any number of `Group` elements.
- A `Group` represents a parameter group and can contain any number of `Key` elements.
- The attribute `name` within a `Group` element defines the name of the parameter group.

- `Key` represents a parameter and has the following attributes and contents:
  - `name`: Parameter name
  - `type`: Parameter type `String`, `Boolean`, `Integer`, `Decimal` or `Date` (ISO date format `YYYY-MM-DD`)
  - Parameter value corresponding to the parameter type

Modelling the Technical Process

1. Create a new Technical Process via the menu **New > Technical Process**.
2. Drag and drop the input XML file into the Technical Process model and select the operation *Read*.
3. Drag and drop the Global Parameter Management adapter into the Technical Process model and select the adapter operation *AddKeys*.
4. Save the Technical Process and run it.
   The parameters will be created and can be viewed and edited both within the Global parameters and groups and the Administration interface within X4 Control Center.

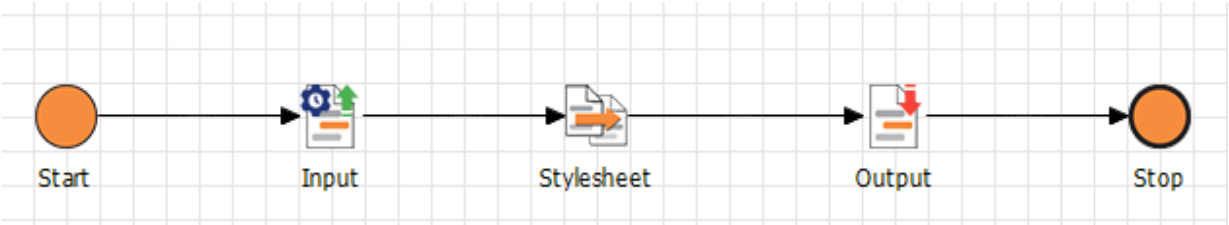
Using Global Parameters in XSL Mappings

As with system and project parameters, global parameters with `%PARAM_Group_Name%` can also be used in XSL stylesheets.

Example

Using global parameters in XSL mappings

The following example shows the use of global parameters in mappings. Before using them, the relevant parameters must be inserted in the XSL mapping.



Creating the XSL mapping

The XSL mapping must have the following structure. The global parameters to be used have to be declared within the stylesheet with their full qualifies names, first. Then, they can be used as usual.

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>
```

Modelling the Technical Process

1. Create a new Technical Process via the menu **New > Technical Process**.
2. Drag and drop the XSL mapping (see example above) into the Technical Process model.
3. Save the Technical Process and run it.
   The parameters will be loaded and can now be used.

5.6.3 Project Parameters

How can I use project parameters?

In an existing project, you can create the **Project Parameter Configuration**, a file with the extension `*.param`, where you can add project-specific parameters and parameter groups.

In addition, you can declare **Custom Types** (Enums) with **Custom Type Values** there, which you can select later when creating a project parameter. A possible use case of **Custom Types** with **Custom Type Values** is the definition of conditions in the **Condition Editor** (see Creating and Using Conditions).

Project parameters can then be used just like global and system parameters in the X4 Designer (in the editors of adapters, mappings, processes, conditions and the **Properties** view) and can be exported or imported with the project.

Project parameters and project parameter groups are loaded at server startup and deployed with the respective project.

They can be modified and persisted at runtime via ReST API, X4 Adapter or Web Apps.

Benefits of project parameters

- Reusability of components through parameterization is increased.
- The process can be influenced directly at runtime by project parameters.
- With the `*.param` file, project parameters can be exported or imported directly with the project.
- Via the X4 Control Center (Swagger UI) or via a special adapter, the Project Parameter Management adapter, project parameters can be changed at runtime
- Project parameters, just like system and global parameters, can be used in mappings, conditions and other process components, such as adapters, as well as in business rules or human tasks (BPM projects).
- Global parameters can be copied to the **Project Parameter Configuration** in order to be used there in a project-specific way (see Copying global parameters to project parameters).

Special features of project parameters

- **Scoping**: A parameter value can be shared by all processes or be different for each process execution.
- **Password encryption**: Project parameters of **Type** `Password` are encrypted by a runtime mechanism. No passwords should be configured in the file at development time.
- **Read only**: A parameter can be defined as read-only, i.e. as a constant, so that it cannot be changed at runtime.
- **Access Restriction**: You can specify that a parameter can only be edited by users with a specific role in the authentication provider Keycloak (see https://www.keycloak.org in the Docs> Server Administration section).
- **Metadata**: Metadata (i.e. description) provides information about the parameter for configuration purposes

- **X4 Control Center**: The X4 Control Center (Swagger UI) can be used to view and adjust all parameters that the currently logged in user is allowed to see or edit.

Additional information

- Creating Project Parameters
- Renaming Project Parameters
- Creating custom types
- Reusing Global and System Parameters

5.6.3.1 Managing Project Parameters via the Adapter Project Parameter Management

The adapter Project Parameter Management can be used to create and manage project parameters. The adapter offers various operations for managing project parameters, such as adding or deleting, but also importing and exporting parameters or parameter groups.

5.6.3.2 Controlling Project Parameters in the X4 Control Center

During operation, administrators can view and adjust the project parameter configuration in the X4 Control Center at any time via Administration > X4 API (Swagger UI). New process instances are then executed with modified parameters.

5.6.3.3 Creating Project Parameters

Project parameters and project parameter groups can be created in a project-specific file with the extension `*.param` (**Project Parameter Configuration**).

Project parameters can then be used just like global and system parameters in the X4 Designer (in the editors of adapters, mappings, processes, conditions, and the **Properties** view) and can be exported or imported with the project.

Project parameters and project parameter groups are loaded at server startup and deployed with the respective project.

They can be changed and persisted at runtime via ReST API, X4 adapters or Web Apps.

Note

Observe the following prioritization when using parameters:
System parameters → Project parameters → Global parameters.
In case of name conflicts, an error message or warning is displayed in the **Problems** view:

- Error message in case of name conflicts of system and project parameters
- Warning in case of name conflicts of project parameters and global parameters

Note

With the shortcut `Ctrl+Space` all available parameters (global, system and project parameters) are displayed in process components, mappings, adapters, and conditions.

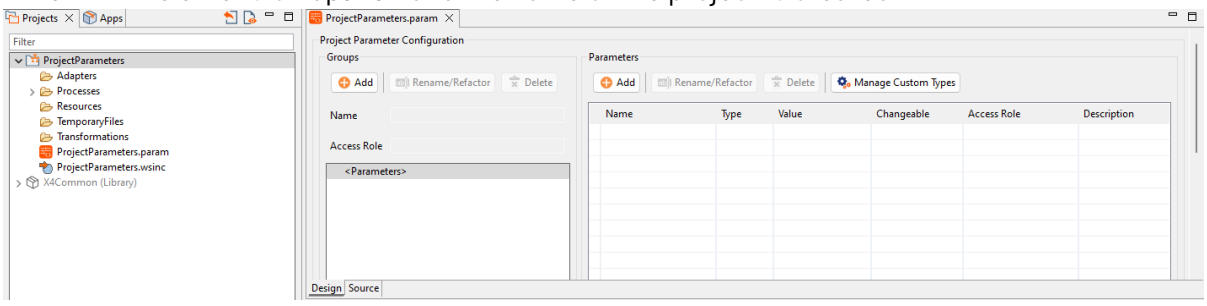
Content of this page:

- Creating the Project Parameter Configuration
- Creating Project Parameters
- Creating Project Parameter Groups
- Example of Project Parameter Configuration
  - Project Parameter Configuration with Constant and Variable at Root Level
  - Project Parameter Configuration with Project Parameter Group
  - Example of an XML Structure

Creating the Project Parameter Configuration

How to create the Project Parameter Configuration

1. Select an existing ESB project and right-click the project folder to open the context menu.
2. In the context menu, select **New** > **Project Parameter Configuration**
   A file with the extension `.param` and the name of the project is created:

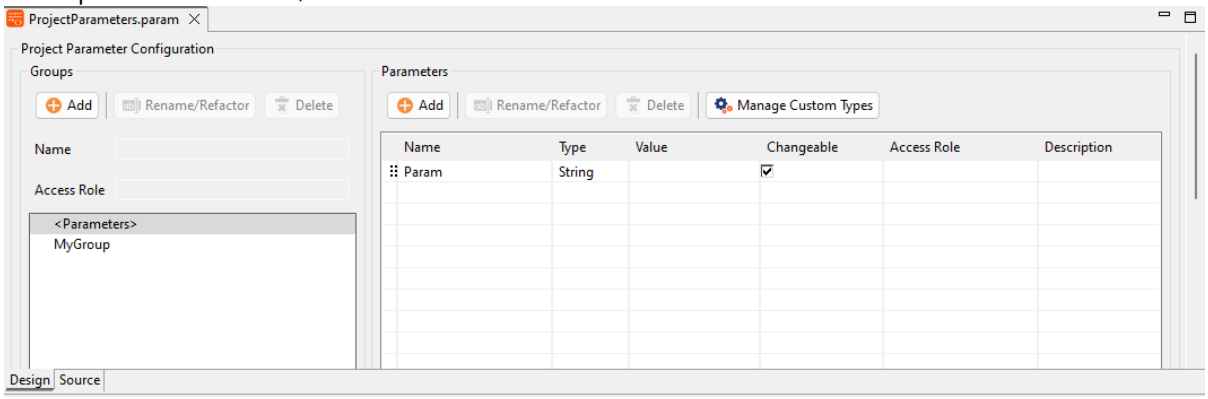


Creating Project Parameters

> **Note**
>
> - Editing of project parameters and project parameter groups can be restricted by roles in the authentication provider Keycloak (see https://www.keycloak.org in the section **Docs > Server Administration**).
> - Parameters can be added in the Designer in the **Project Parameter Configuration** via the **Design** view (recommended) or directly in the XML structure via **Source** view.
> - Custom Types (Enums) can be created via **Manage Custom Types**. Here, for example, values for conditions can be created and then used in the **Condition Editor** (see Manage Custom Types).

How to create project parameters

1. Open the **Project Parameter Configuration**.
2. In the pane **Parameters**, click **Add**.



3. Enter the properties for the project parameter. Data fields with * are mandatory fields:

Data field	Description
*Name**	Name of the parameter **Note** To avoid name conflicts, the name of the parameter must be unique. **Note** The name of the project parameter must start with a letter, must not contain special characters and dots. The following characters are allowed: `^[A-Za-z_][A-Za-z0-9-_]*$` **Valid example**: `My-Project-Parameter-123`
*Type**	Data type of the parameter **Note** The type of the parameter is checked when it is created and when it is overwritten at runtime. Based on the parameter type, the parameter values are used in the project files. Possible values: • String • Boolean • Integer • Decimal • Date • DateTime • Password **Note** Password type parameters are displayed in the **Design** view with asterisk and in the **Source** view (XML structure) encrypted. • **Custom Types** with **Custom Type Values** (Enums)
*Value**	Default value of the parameter

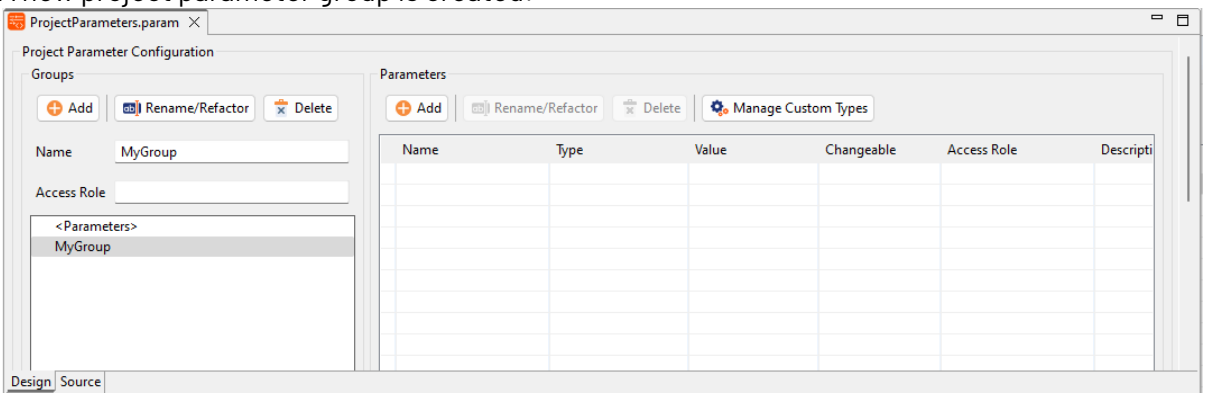
Data field	Description
*Changeable**	Type of the parameter: **Constant** or **Variable** **Note** Parameters of type **Constant** cannot be overwritten at runtime. Project parameters of type **Variable** can be overwritten at runtime. Thus, the value does not have to be filled for project parameters of the type **Variable**. Possible values (Boolean): • True (Variable) • False (Constant)
Access Role	Role with access rights to edit the parameter
Description	Description of the parameter

Creating Project Parameter Groups

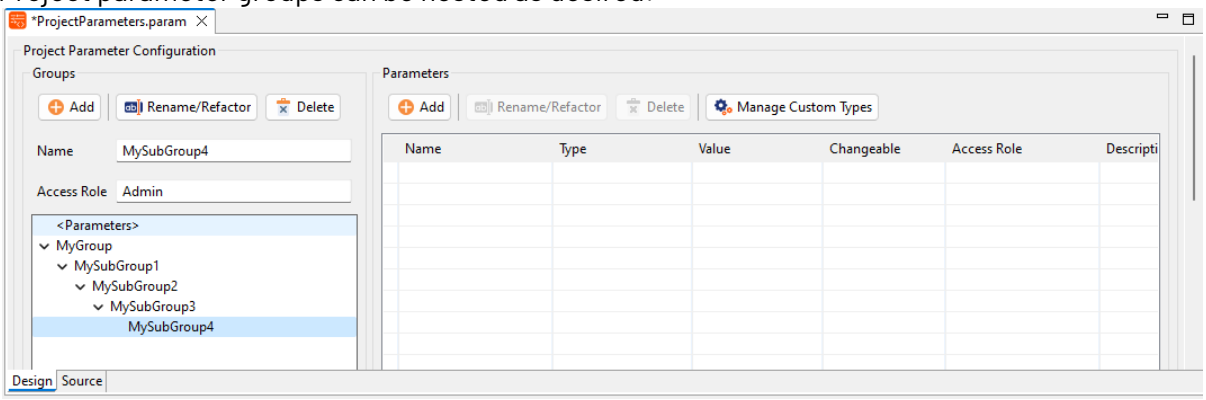
Note
To avoid name conflicts, the name of the parameter must be unique.

How to create a project parameter group

1. Open the **Project Parameter Configuration**.
2. In the pane `Groups`, click **Add**.
   A new project parameter group is created:

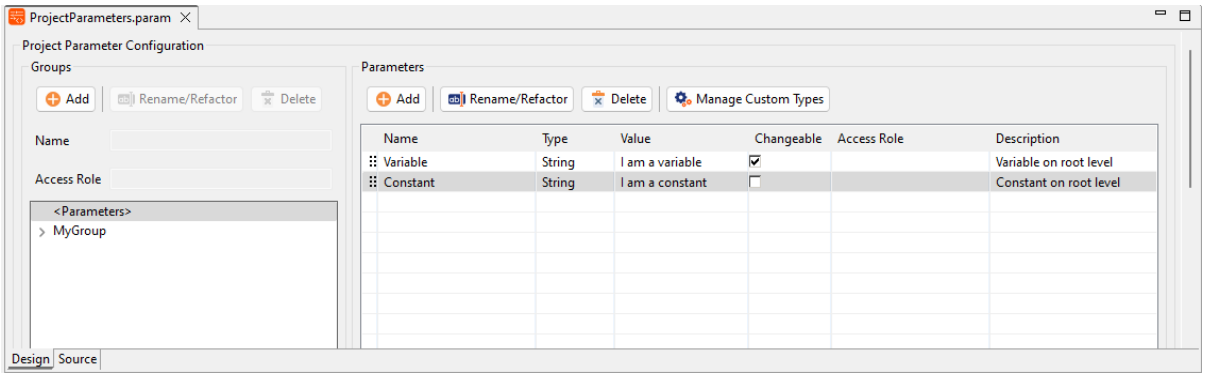


Project parameter groups can be nested as desired:

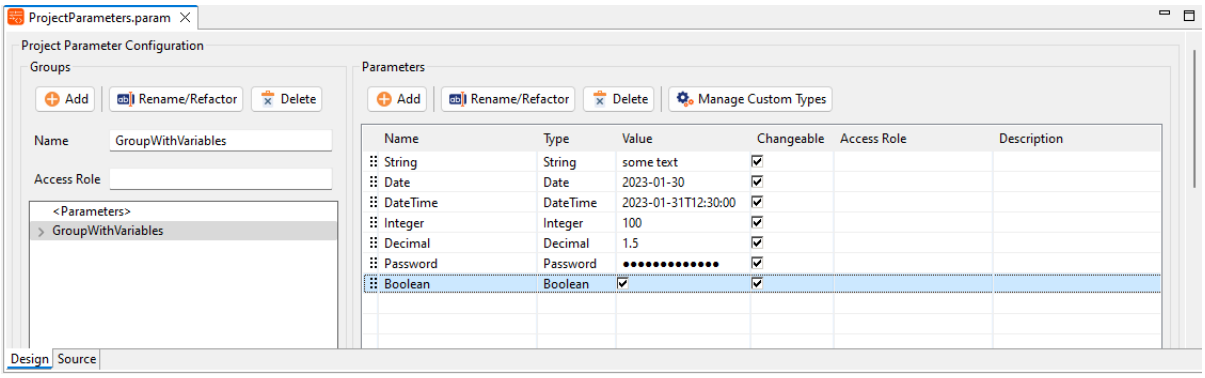


Example of Project Parameter Configuration

Project Parameter Configuration with Constant and Variable at Root Level



Project Parameter Configuration with Project Parameter Group



Example of an XML Structure

Example of XML structure of a **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 Renaming Project Parameters

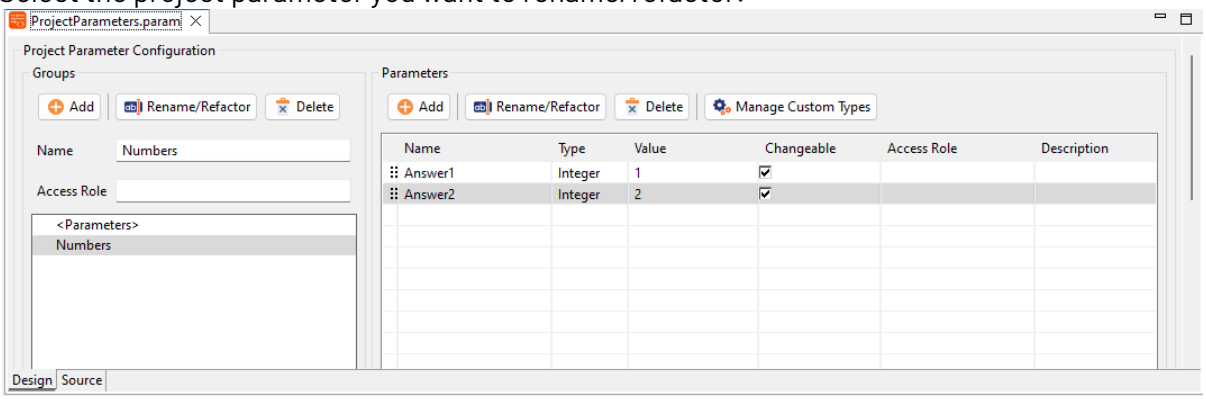
The **Rename**/**Refactor** button allows you to rename project parameters and groups of project parameters. This renaming is applied by referencing (refactoring) in all project files in which the project parameters or groups of project parameters are used.

> **Note**
> If only the name of the project parameter is changed in the **Name** column, no refactoring takes place, i.e. the project parameter is not renamed in the files in the project. Refactoring can only take place via the **Rename**/**Refactor** button.

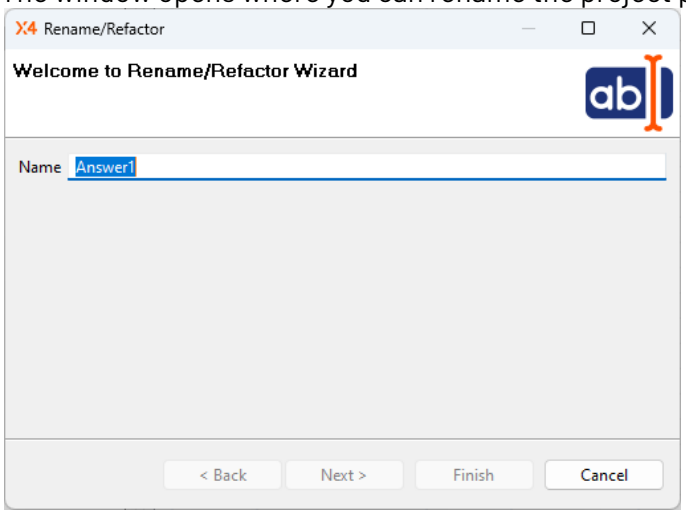
How to rename project parameters

> **Note**
> To rename/refactor a group, follow the same steps.

1. Open an existing `*.param` file (**Project Parameter Configuration**).
2. Select the project parameter you want to rename/refactor:

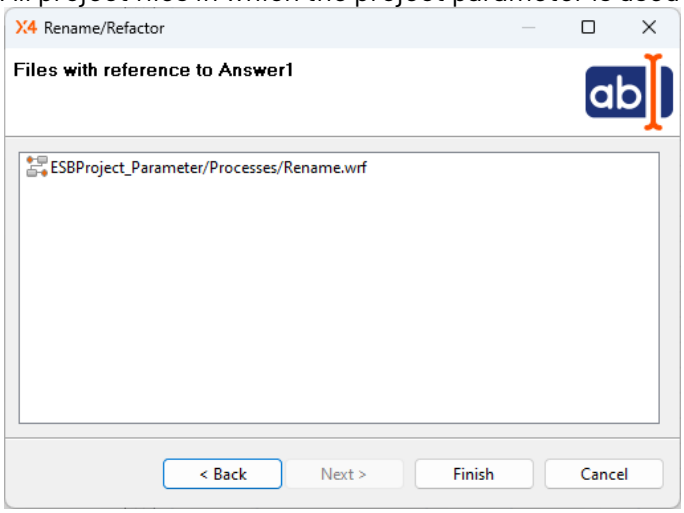


3. Click **Rename/Refactor**.
   The window opens where you can rename the project parameter:



4. In the window **Rename/Refactor**, enter a new name for the project parameter.

5. Click **Next**.
   All project files in which the project parameter is used are displayed:



6. Click **Finish** to rename/refactor the project parameter in all places used in the project.
   The project parameter has been renamed/refactored in all files in the project.

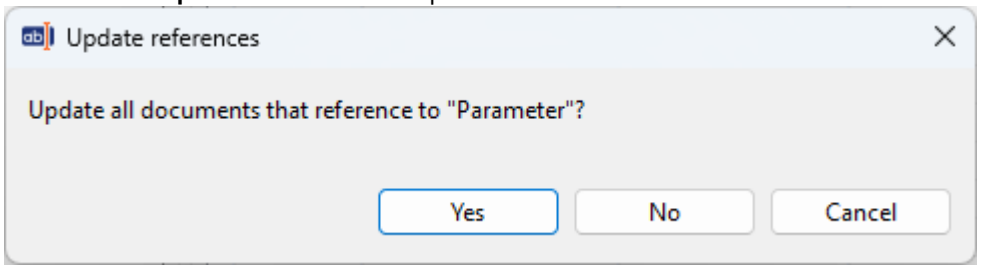
Moving Project Parameters to Another Project Parameter Group

In the **Project Parameter Configuration,** you can move one or more project parameters to another group via drag&drop and optionally re-reference them.

A window opens with the query whether a refactoring of the project parameter should take place or not.

How to move project parameters to another group

1. Open an existing `*.param` file (**Project Parameter Configuration**).
2. Drag and drop the project parameter you want to move to the desired group.
3. The window **Update References** opens:



4. Confirm with **Yes** to perform a refactoring for the selected project parameter. During refactoring, the selected project parameter is renamed and newly referenced.
   The project parameter has been moved to the group and renamed in all files (processes, mappings, conditions, adapters) in the project.

5.6.3.5 Creating custom types

In the **Project Parameter Configuration**, you can define **Custom Types** (Enums) with **Custom Type Values** (Enum values) that you can select later when creating a project parameter.

Custom Types are created via **Manage Custom Types**.

Possible use case:

- **Custom Type Values** as selection options in the **Condition Editor** (see Creating and using conditions).

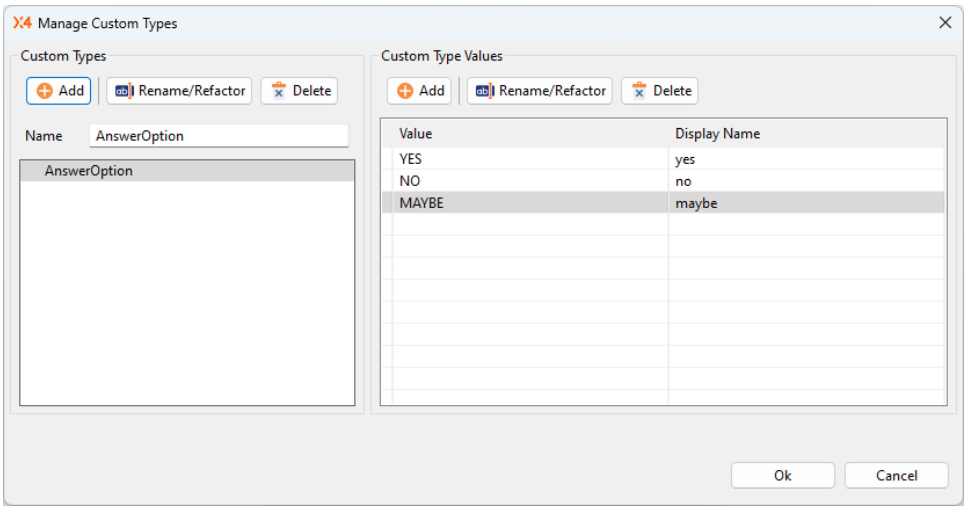
Content of this page:

- Creating Custom Types with Custom Type Values
- Deleting Custom Types and Custom Type Values
  - Deleting a Custom Type
  - Deleting Custom Type Values
- Renaming Project Parameters with Custom Types

Creating Custom Types with Custom Type Values

Use case:

In this example, we create the **Custom Type** `AnswerOption` with the **Custom Type Values** `YES`, `NO`, `MAYBE` and the **Display Names** `yes`, `no`, `maybe`:

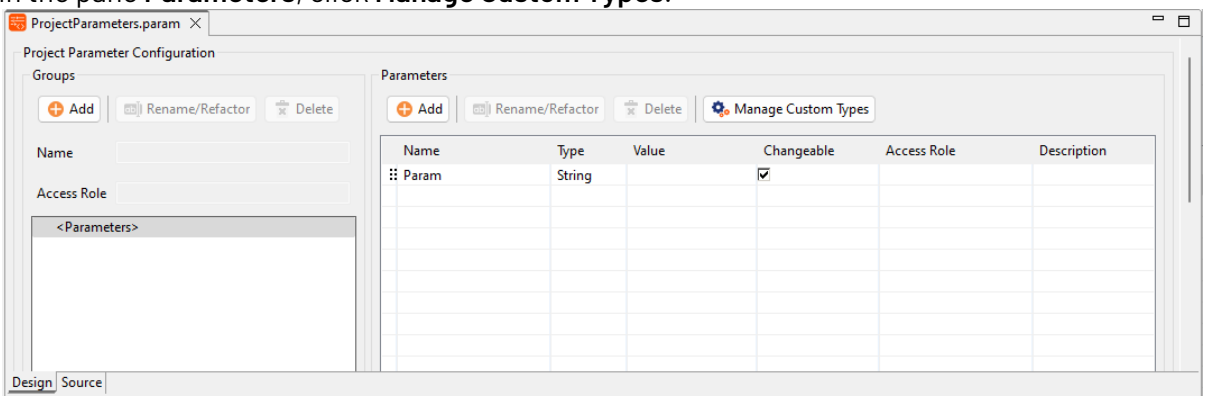


How to open the window Manage Custom Types

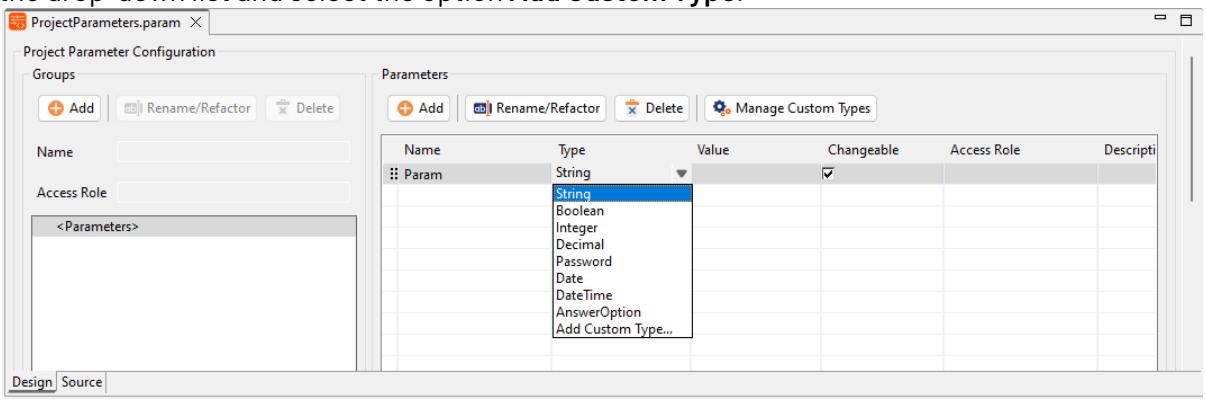
Open an existing file `*.param` (**Project Parameter Configuration**).

There are two ways how you can add **Custom Types**:

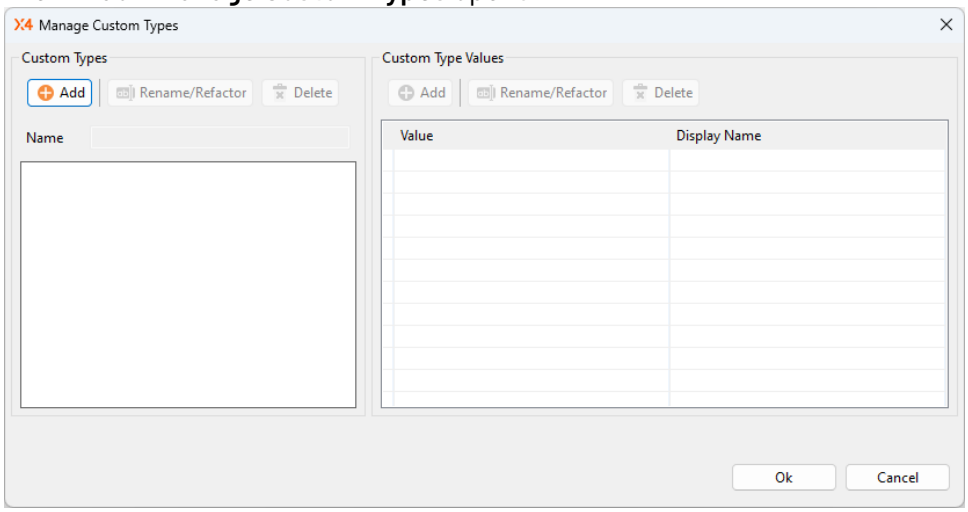
1. In the pane **Parameters**, click **Manage Custom Types**.



2. In the pane **Parameters**, in the **Type** column of the new project parameter to be created, open the drop-down list and select the option **Add Custom Type**.



The window **Manage Custom Types** opens:



How to create Custom Types with Custom Type Values

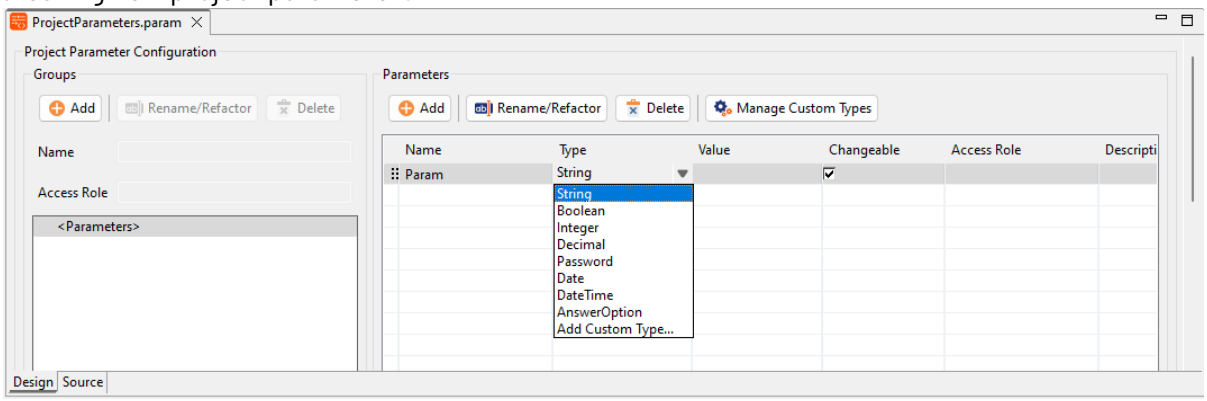
1. In the pane **Custom Types**, click **Add** to add a new **Custom Type**.
   A new **Custom Type** with the predefined name **CustomType** has been created.

> **Note**
> The name of the **Custom Type** must be unique.

In the pane **Custom Type Values**, you can now add the **Custom Type Values** with **Display Name**.

2. Click **Add** to add a new **Custom Type Value** with **Display Name**.
3. Click **OK** to save the **Custom Types** with the defined values.
   A new **Custom Type** with **Custom Type Values** has been created.

The new **Custom Type** with **Custom Type Values** can now be selected in the **Type** column when creating new project parameters:



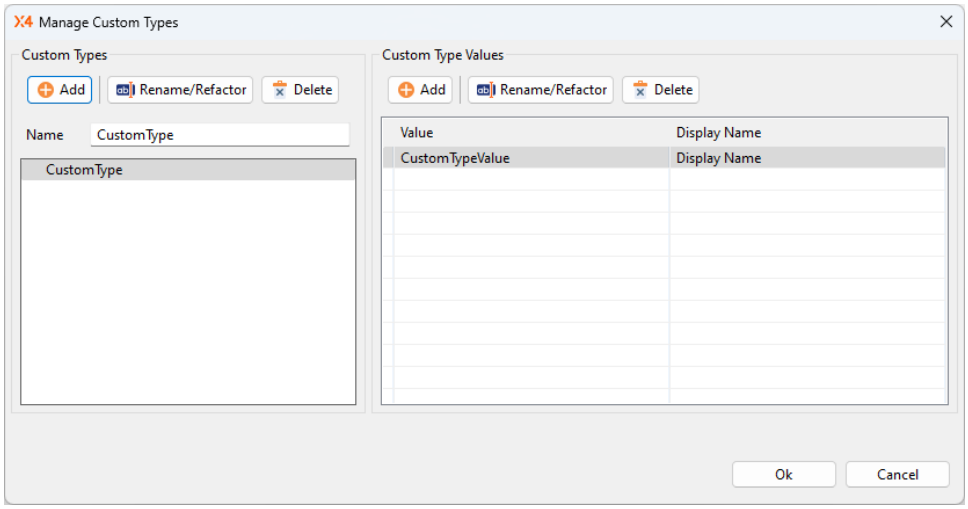
Field	Description	Name in Source (XML)
Custom Type Name	Name of the **Custom Type**, selectable in the column **Type** as additional **Type**	`Enum name`
Custom Type Value – Value	Selection options (**Values**), selectable in the column **Value** when creating a new project parameter with **Custom Type**	`<Value>yes <Value>`
Attribute: Custom Type Value – Display Name	**Display name** of the **value**	`displayName` Attribute: 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>
```

How to change the name of a Custom Type

> **Note**
> You can change the name of a **Custom Type** in the pane **Custom Types** in the field **Name**.

1. Open the window **Manage Custom Types**.
2. In the pane **Custom Types**, click on the **Custom Type**.
3. Edit the name in the field **Name**:

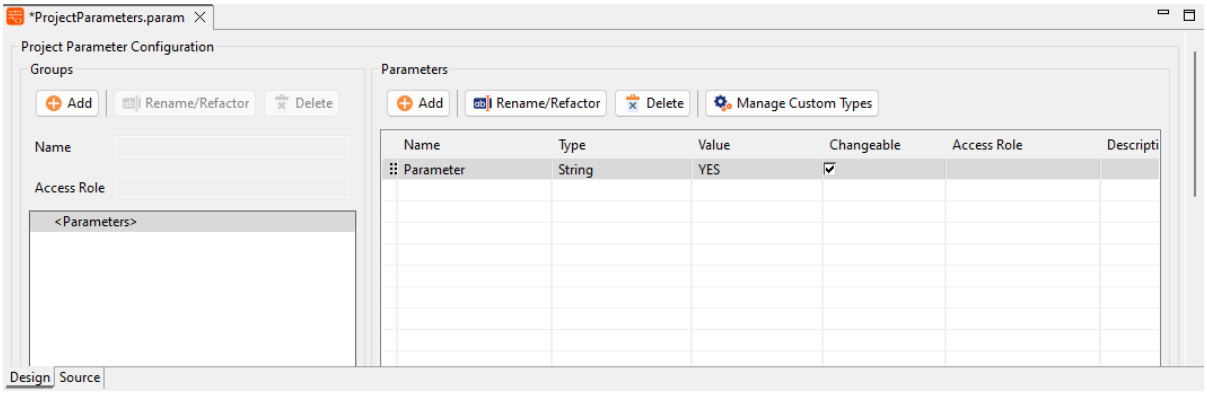


Deleting Custom Types and Custom Type Values

Deleting a Custom Type

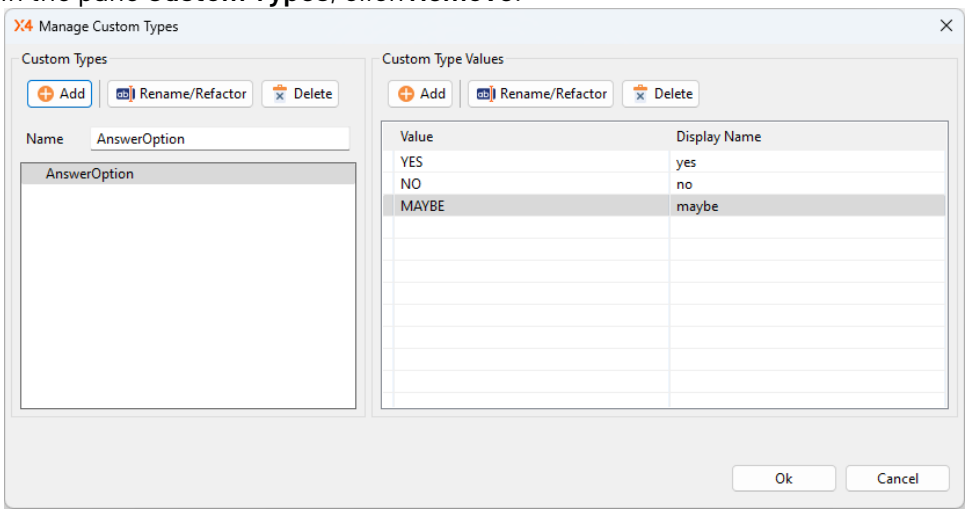
What happens when I delete a **Custom Type** that is in use?

When a **Custom Type** is deleted, the **Type** is set to `String`, but the previously set **Custom Type Value** in the **Value** column is retained:

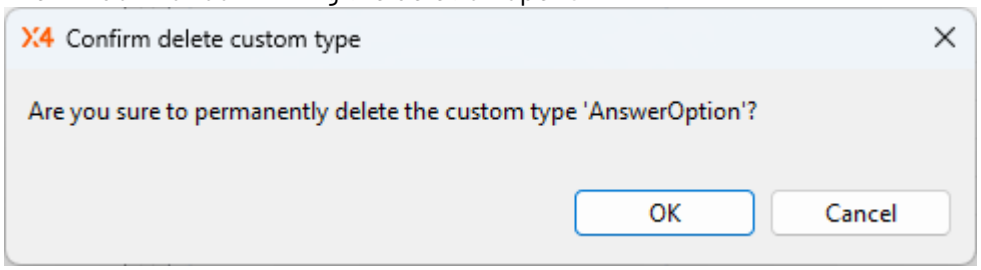


How to delete a Custom Type:

1. Open an existing `*.param` file (**Project Parameter Configuration**).
2. Click **Manage Custom Types**.
3. In the pane **Custom Types**, select the **Custom Type** you want to delete.
4. In the pane **Custom Types**, click **Remove**.



5. The window for confirming the deletion opens:

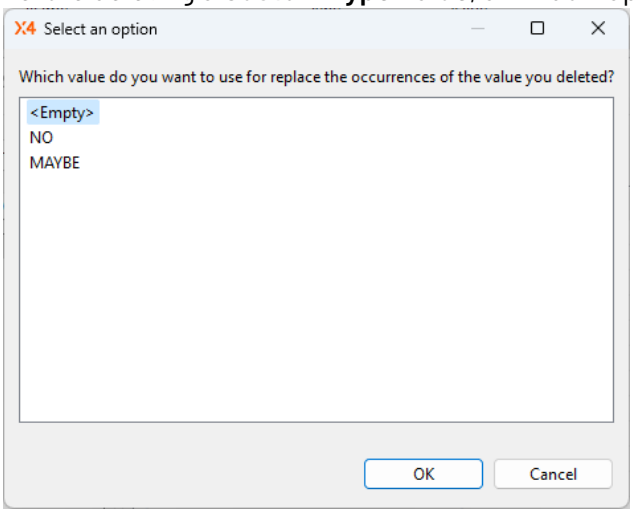


6. Click **OK** to confirm the deletion of the **Custom Type**.
   The **Custom Type** is deleted.

Deleting Custom Type Values

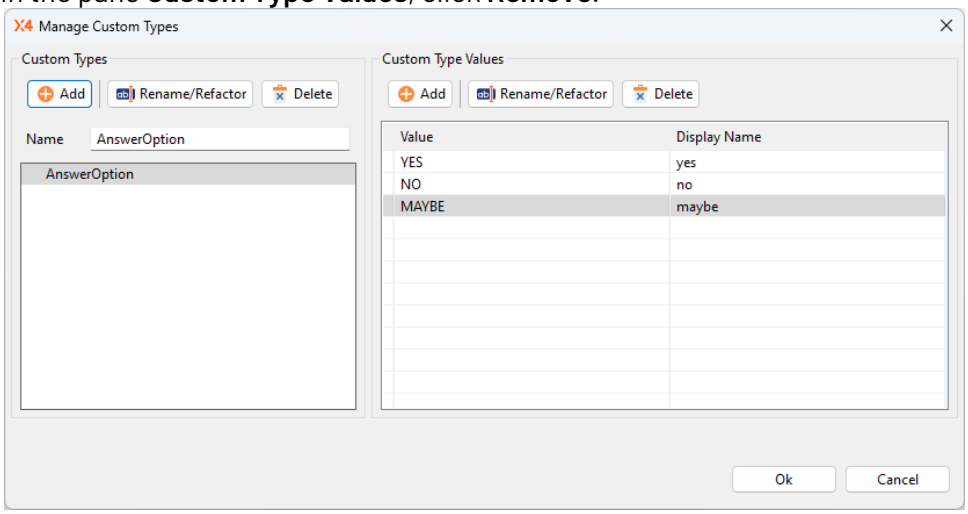
What happens if I delete a **Custom Type Value** that is in use?

Before deleting a **Custom Type Value**, a window appears asking which value should be set instead.

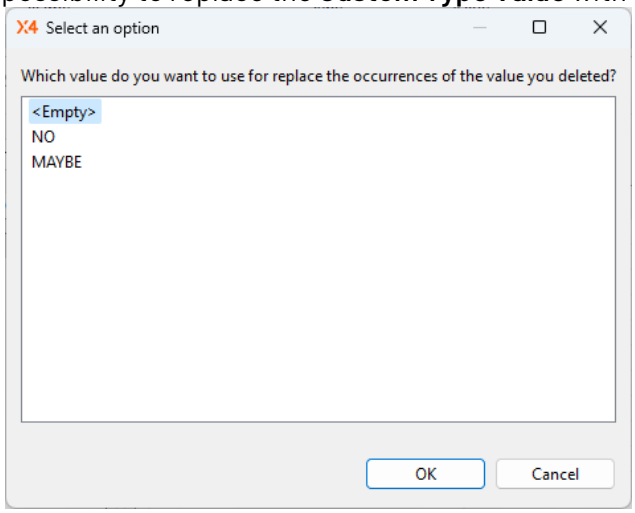


How to delete a Custom Type Value:

1. Open an existing `*.param` file (**Project Parameter Configuration**).
2. Click **Manage Custom Types**.
3. In the pane **Custom Type Values**, select the **Custom Type Value** you want to delete.
4. In the pane **Custom Type Values**, click **Remove**.



5. The window for confirming the deletion opens. If the **Custom Type Value** is in use, you have the possibility to replace the **Custom Type Value** with another one:



6. Select the value to be used for replacement.
7. Click **OK** to confirm the deletion of the **Custom Type Value**.
8. The **Custom Type Value** has been deleted, and the value selected for replacement has been used.

Renaming Project Parameters with Custom Types

Renaming project parameters with **Custom Types** is done like renaming project parameters (see Renaming project parameters).

5.6.3.6 Reusing Global and System Parameters

Global and system parameters can be reused in the **Project Parameter Configuration** `*.param`.

When does it make sense to reuse global and system parameters?

Reuse is useful so that parameters do not have to be redefined over and over again, but existing parameters can be reused, e.g. in **Mappings** or as parameters or **Component Parameters** in an adapter.

> **Note**
> To avoid naming conflicts, the name of parameters and parameter groups must be unique within a level (at root level and for nested groups).

> **Note**
> The following should be noted when reusing parameters in the **Project Parameter Configuration** `*.param`:
>
> - constants may only be reused in other constants
> - global, system or project parameters may be reused in variables
> - parameters must not contain cycles (e.g. variable b uses the value of variable a and variable a uses the value of variable b)
> - only parameters of type String may be reused

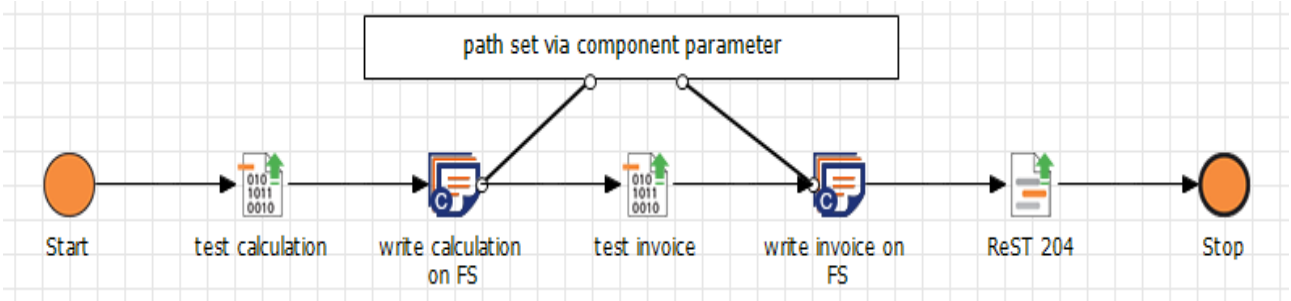
Possible Use Case

In the **File System Adapter** a path can be set via **Component Parameters**. A destination folder with file name can be referenced here.

> **Note**
> Instead of a **Component Parameter**, another type of parameterization can also be selected (e.g. setting parameters directly on the adapter, see also Parameterization Options).

Example

In this process, one file at a time is written to a destination folder. The destination folder and the file name are defined via **Component Parameters** in the **File System Adapter**. Global, system and project parameters are reused in the **Component Parameters**.



Component Parameter in **File System Adapter** *Write Calculation on FS*:

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

Component Parameter in **File System Adapter** *Write Invoice on FS*:

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

In the project parameter `%fileRootPath%`, the following parameters are reused: project parameter (constant) `%X4tempfileDirectory%` and system parameter `%CURRENT_PROJECT%`.

In the project parameter `%timestampedFile%`, the following parameters are reused: the project parameters `%filePrefix%` and `%textfileEnding%` as well as the system parameter `%NOW%`.

Example of an XML structure with reused parameters in the **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 Help Functions

Help functions that are provided by the X4 BPMS to be used within XSL mappings are introduced in this chapter.

5.7.1 TimeHelper (deprecated)

> ⚠ The date and time functions of the Java class `de.softproject.xsl.TimeHelperDate` provided within XSL mappings are deprecated and will be removed in future versions of the X4 BPMS.

The Java class `de.softproject.xsl.TimeHelperDate` is located within the X4 Server library `x4-client.jar` (under `<X4>\Server\<wildfly>\modules\system\layers\base\de\softproject\x4\client\main`).

> ⓘ An XSL mapping using this help class, can't be executed in the Mapping Editor's transformation preview.

5.7.1.1 Date/Time Methods

addDay(*java.util.Date* date, *int* amount**)**	Adds any integer number of days to a `java.util.Date` object and returns a `static java.util.Date` object
addHour(*java.util.Date* date, *int* amount**)**	Adds any integer number of hours to a `java.util.Date` object and returns a `static java.util.Date` object
addMillisecond(*java.util.Date* date, *int* amount**)**	Adds any integer number of millisecond to a `java.util.Date` object and returns a `static java.util.Date` object
addMinute(*java.util.Date* date, *int* amount**)**	Adds any integer number of minutes to a `java.util.Date` object and returns a `static java.util.Date` object
addMonth(*java.util.Date* date, *int* amount)	Adds any integer number of months to a `java.util.Date` object and returns a `static java.util.Date` object
addSecond(*java.util.Date* date, *int* amount**)**	Adds any integer number of seconds to a `java.util.Date` object and returns a `static java.util.Date` object
addYear(*java.util.Date* date, *int* amount**)**	Adds any integer number of years to a `java.util.Date` object and returns a `static java.util.Date` object
asMilliseconds(*java.util.Date* date**)**	Returns a `java.util.Date` object as `static long` number containing the time since 1.1.1970 in milliseconds

canParse(*java.lang.String* format, [*java.lang.String* timezone,] *java.lang.String* value**)**	Checks the format of a date with the optional parameter `timezone` for the timezone and returns a `static boolean` object ? Unknown Attachment In this case, only the format and not the result's validity is checked.
canParseDefaultFormat([*java.lang.String* timezone,] *java.lang.String* value**)**	Checks if the used date format with the optional parameter `timezone` for the timezone corresponds to the default format and returns a `static boolean` object
createDefaultFormat([*java.lang.String* timezone]**)**	Outputs the current date (with the optional parameter `timezone` for a specific timezone) in the default format `static java.text.SimpleDateFormat`
createFormat(*java.lang.String* format, [*java.lang.String* timezone]**)**	Outputs the current date (with the optional parameter `timezone` for a specific timezone) in the format `static java.text.SimpleDateFormat` ? Unknown Attachment For more information regarding the date and time formats, consult http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html
differenceInMilliseconds(*java.util.Date* from, *java.util.Date* to**)**	Calculates the difference between two `java.util.Date` objects in milliseconds and returns a `static long` number
format(*java.lang.String* format, [*java.lang.String* timezone,] *java.util.Date* date**)**	Formats a `java.util.Date` object with optional parameter `timezone` for the timezone into a `static java.lang.String` object
formatDefaultFormat([*java.lang.String* timezone,] *java.util.Date* date**)**	Formats a `java.util.Date` object with optional parameter `timezone` for the timezone into an object with the default format `static java.lang.String`
fromMilliseconds(*long* milli**)**	Converts a date in milliseconds into a `static java.util.Date` object
getDay(*java.util.Date* date, [*java.lang.String* timezone]**)**	Determines the day from a `java.util.Date` object with optional parameter `timezone` for the timezone, and outputs it as `static int` object

getDayOfWeek(*java.util.Date* date, [*java.lang.String* timezone]**)**	Determines the weekday from a `java.util.Date` object with optional parameter `timezone` for the timezone, and outputs it as `static int` object ? Unknown Attachment The counting of the weekdays begins on Sunday with 1: • Sunday: *1* • Monday: *2* • Tuesday: *3* • Wednesday: *4* • Thursday: *5* • Friday: *6* • Saturday: *7*
getDayOfWeekInMonth(*java.util.Date* date, [*java.lang.String* timezone]**)**	Determines the weekday within a month (e. g. which Tuesday of the month) from a `java.util.Date` object with optional parameter `timezone` for the timezone, and outputs it as integer number of the type `static int`
getDayOfYear(*java.util.Date* date, [*java.lang.String* timezone]**)**	Determines the day within a year from a `java.util.Date` object with optional parameter `timezone` for the timezone, and outputs it as integer number of the type `static int`
getHour(*java.util.Date* date, [*java.lang.String* timezone]**)**	Determines the hours from a `java.util.Date` object with optional parameter `timezone` for the timezone, and outputs it as integer number of the type `static int`
getMilliSecond(*java.util.Date* date, [*java.lang.String* timezone]**)**	Determines the milliseconds from a `java.util.Date` object with optional parameter `timezone` for the timezone, and outputs it as integer number of the type `static int`
getMinute(*java.util.Date* date, [*java.lang.String* timezone]**)**	Determines the minutes from a `java.util.Date` object with optional parameter `timezone` for the timezone, and outputs it as integer number of the type `static int`

getMonth(*java.util.Date* date, [*java.lang.String* timezone]**)**	Determines the month from a `java.util.Date` object wit h optional parameter `timezone` for the timezone, and outputs it as integer number of the type `static int` ? Unknown Attachment The counting of the months begins with 0: • January: *0* • February: *1* • March: *2* • April: *3* • May: *4* • June: *5* • July: *6* • August: *7* • September: *8* • October: *9* • November: *10* • December: *11*
getSecond(*java.util.Date* date, [*java.lang.String* timezone]**)**	Determines the seconds from a `java.util.Date` object wi th optional parameter `timezone` for the timezone, and outputs it as integer number of the type `static int`
getWeekOfMonth(*java.util.Date* date, [*java.lang.String* timezone]**)**	Determines the week of the month from a `java.util.Date` object with optional parameter `timezon e` for the timezone, and outputs it as integer number of the type `static int`
getWeekOfYear(*java.util.Date* date, [*java.lang.String* timezone]**)**	Determines the week number from a `java.util.Date` obj ect with optional parameter `timezone` for the timezone, and outputs it as integer number of the type `static int`
getYear(*java.util.Date* date, [*java.lang.String* timezone]**)**	Determines the year from a `java.util.Date` object with optional parameter `timezone` for the timezone, and outputs it as integer number of the type `static int`
isAfter(*java.util.Date* date, *java.util.Date* reference**)**	Checks if a `java.util.Date` object lies before a reference object of the same type, and outputs a `boolean` o bject
isBefore(*java.util.Date* date, *java.util.Date* reference**)**	Checks if a `java.util.Date` object lies after a reference object of the same type, and outputs a `boolean` object
now()	Determines the current date and outputs it as `static java.util.Date` object

parse(*java.lang.String format,* [*java.lang.String* timezone,] *java.lang.String* value**)**	Reads a date in `java.lang.String` format with optional parameter `timezone` for the timezone, and returns a `static java.util.Date` object
parseDefaultFormat([*java.lang.String* timezone,] *java.lang.String* value**)**	Reads a date in `java.lang.String` format with optional parameter `timezone` for the timezone, and returns a `static java.util.Date` object in the default format
setBeginOfDay(*java.util.Date* date, [*java.lang.String* timezone]**)**	Outputs for the entered `java.util.Date` object (with optional parameter `timezone` for the timezone) the beginning of the day as `static java.util.Date` object
setBeginOfMonth(*java.util.Date* date, [*java.lang.String* timezone]**)**	Outputs for the entered `java.util.Date` object (with optional parameter `timezone` for the timezone) the beginning of the month as `static java.util.Date` object
setBeginOfYear(*java.util.Date* date, [*java.lang.String* timezone]**)**	Outputs for the entered `java.util.Date` object (with optional parameter `timezone` for the timezone) the beginning of the year as `static java.util.Date` object
setDay(*java.util.Date* date, [*java.lang.String* timezone], *int* amount**)**	Outputs for the entered `java.util.Date` object (with optional parameter `timezone` for the timezone) the day of the month as `static java.util.Date` object ? Unknown Attachment If the entered number is higher than the actual number of days within the respective month, the difference will be added to the following month(s).
setDayOfYear(*java.util.Date* date, [*java.lang.String* timezone], *int* amount**)**	Outputs for the entered `java.util.Date` object with optional parameter `timezone`, the timezone of the date as `static java.util.Date` object
setEndOfDay(*java.util.Date* date, [*java.lang.String* timezone]**)**	Outputs for the entered `java.util.Date` object (with optional parameter `timezone` for the timezone) the end of the day as `static java.util.Date` object
setEndOfMonth(*java.util.Date* date, [*java.lang.String* timezone]**)**	Outputs for the entered `java.util.Date` object (with optional parameter `timezone` for the timezone) the end of the month as `static java.util.Date` object
setEndOfYear(*java.util.Date* date, [*java.lang.String* timezone]**)**	Outputs for the entered `java.util.Date` object (with optional parameter `timezone` for the timezone) the end of the year as `static java.util.Date` object

setHour(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Outputs for the entered `java.util.Date` object (with optional parameter `timezone` for the timezone) the number of hours as `static java.util.Date` object ? Unknown Attachment If the entered number is higher than 24, the difference will be added to the following day(s).
setMillisecond(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Outputs for the entered `java.util.Date` object (with optional parameter `timezone` for the timezone) the number of milliseconds as `static java.util.Date` object
setMinute(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Outputs for the entered `java.util.Date` object (with optional parameter `timezone` for the timezone) the number of minutes as `static java.util.Date` object ? Unknown Attachment If the entered number is higher than 60, the difference will be added to the following hour(s).
setMonth(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Outputs for the entered `java.util.Date` object with optional parameter `timezone` for the timezone the month as `static java.util.Date` object ? Unknown Attachment The counting of the months begins with 0: • January: *0* • February: *1* • March: *2* • April: *3* • May: *4* • June: *5* • July: *6* • August: *7* • September: *8* • October: *9* • November: *10* • December: *11* If the entered number is higher than 11, the difference will be added to the following year(s).

setSecond(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Outputs for the entered `java.util.Date` object with optional parameter `timezone` for the timezone the seconds as `static java.util.Date` object ? Unknown Attachment If the entered number is higher than 60, the difference will be added to the following minute(s).
setYear(*java.util.Date* date, [*java.lang.String* timezone,] *int* amount**)**	Outputs for the entered `java.util.Date` object with optional parameter `timezone` for the timezone the year as `static java.util.Date` object

5.7.1.2 Example

Representative for all methods, in this example variables with one method call are initialized and their return values are outputted in an XML element. For this purpose, the two namespaces `x4time` for the time helper class and `sdf` for the Java SimpleDateFormat are defined in the XSL stylesheet's head first. Moreover, in the variables `date` and `from_date` two variables with a `java.util.Date` object are initialized as sample data for future use.

```
<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 (deprecated)

> ⚠ The working time functions of the Java class `de.softproject.xsl.WorkingTimeCalculator` are deprecated and will be removed in future versions of the X4 BPMS.
> Beginning with version 6.0.0 of the X4 BPMS, the Working Time Calculator adapter is available for this purpose.

The Java class `de.softproject.xsl.WorkingTimeCalculator` is located within the X4 Server library `x4-client.jar` (under `<X4>\Server\<wildfly>\modules\system\layers\base\de\softproject\x4\client\main`).

> ⓘ An XSL mapping using this help class, can't be executed in the transformation preview of the Mapping Editor.

5.7.2.1 Methods for Defining the Working Time

Before you can use the methods for the working time calculation, it is required to define the working time (begin and end of the working day, number of working days per week, holidays etc.) for the object `WorkingTimeCalculator` using the following methods.

setHolidays(*java.lang.String* holidays**)**	Sets for a `WorkingTimeCalculator` object one or several holidays as string of the type `java.lang.String` and following the pattern `yyyy-mm-dd, yyyy-mm-dd, ...` and returns a `WorkingTimeCalculator` object. *Example:* `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**)**	Sets for a `WorkingTimeCalculator` object a time period as string of the type `java.lang.String` and following the pattern `yyyy-mm-dd hh:MM-hh:MM, yyyy-mm-dd hh:MM-hh:MM, ...` and returns a `WorkingTimeCalculator` object. This time period is always calculated as working time and has thus priority over holidays and working times. *Example:* `setSpecialDays('2010-12-06 11:30-12:0,2010-01-06 10:30-15:30')`

setWorkingDayBegin(*int* hr, *int* min**)**	Sets for a `WorkingTimeCalculator` object the beginning of a working day including the hours and minutes as integer numbers and returns a `WorkingTimeCalculator` object. *Example:* `setWorkingDayBegin(9,0)`
setWorkingDayBegin(*java.lang.String* hr, *java.lang.String* min**)**	Sets for a `WorkingTimeCalculator` object the beginning of a working day including the hours and minutes as string of the type `java.lang.String` and returns a `WorkingTimeCalculator` object. *Example:* `setWorkingDayBegin('09','00')`
setWorkingDayEnd(*int* hr, *int* min**)**	Sets for a `WorkingTimeCalculator` object the end of a working day including the hours and minutes as integer numbers and returns a `WorkingTimeCalculator` object. *Example:* `setWorkingDayEnd(17,0)`
setWorkingDayEnd(*java.lang.String* hr, *java.lang.String* min**)**	Sets for a `WorkingTimeCalculator` object the end of a working day including the hours and minutes as string of the type `java.lang.String` and returns a `WorkingTimeCalculator` object. *Example:* `setWorkingDayEnd('17','00')`
setWorkingDays(*java.lang.String* days**)**	Sets for a `WorkingTimeCalculator` object a list of the working days per week as string of the type `java.lang.String` and following the pattern `<day number>,<day number>,...` and returns a `WorkingTimeCalculator` object. Weekdays are outputted according to `java.util.Calendar` as numbered list beginning with `1` for Sunday. *Example:* `setWorkingDays('2,3,4,5,6')`

5.7.2.2 Working Time Calculating Functions

addWorkingTimeMinutes(*WorkingTimeCalculator* wtc, *java.util.Date* from, *int* minutes**)**	Adds to a `java.util.Date` object any integer number of minutes within the defined working time and outputs a `static java.util.Date` object. *Prerequisite:* a `WorkingTimeCalculator` object has already been initialized correctly and handed over as first argument within an XSL variable, see Methods for defining the working time

addWorkingTimeMinutesString(*WorkingTimeCalculator* wtc, *java.util.Date* from, *int* minutes)	Adds to a `java.util.Date` object any integer number of minutes within the defined working time and outputs a string of the type `java.lang.String`. *Prerequisite:* a `WorkingTimeCalculator` object has already been initialized correctly and handed over as first argument within an XSL variable, see Methods for defining the working time ? Unknown Attachment Only compatible with *Xalan* 2.7.1, but not with *Saxon* 9!
addWorkingTimeMinutesString(*WorkingTimeCalculator* wtc, *java.lang.String* from, *int* minutes**)**	Adds to a string according to `java.lang.String` any integer number of minutes within the defined working time and outputs a string of the type `java.lang.String`. *Prerequisite:* a `WorkingTimeCalculator` object has already been initialized correctly and handed over as first argument within an XSL variable, see Methods for defining the working time ? Unknown Attachment Only compatible with *Xalan* 2.7.1, but not with *Saxon* 9! *Example:* `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)	Compares two `java.util.Date` objects with each other and outputs the difference in milliseconds as `static long` number. ? Unknown Attachment Only compatible with *Xalan* 2.7.1, but not with *Saxon* 9!
dateDiffWorkingMinutes(*WorkingTimeCalculator* wtc, *java.util.Date* from, *java.util.Date* to)	Compares two `java.util.Date` objects with each other and outputs the difference within the defined working times in milliseconds as `long` number. *Prerequisite:* a `WorkingTimeCalculator` object has already been initialized correctly and handed over as first argument within an XSL variable, see Methods for defining the working time ? Unknown Attachment Only compatible with *Xalan* 2.7.1, but not with *Saxon* 9!

getOstern(*double* year**)**	Determines the Easter date for a year, that has been handed over as four-digit number of the type `double` and returns a string of the type `java.lang.String` *Example:* `getOstern(2014)`
getWorkingTimeMinutes(*WorkingTimeCalculator* wtc, *java.util.Date* from, *java.util.Date* to**)**	Determines the number of minutes within the defined working times, which lie between a start and an end date of the type `java.util.Date`, and returns a number of the type `double`. *Prerequisite:* a `WorkingTimeCalculator` object has already been initialized correctly and handed over as first argument within an XSL variable, see Methods for defining the working time
getWorkingTimeMinutes(*WorkingTimeCalculator* wtc, *int* fromYear, *int* fromMonth, *int* fromDay, *int* fromHr, *int* fromMin**)**	Determines the number of minutes within the defined working times, which lie between a start date (defined as integer number for year, month, day, hour and minute) and the current date, and returns a number of the type `double`. *Prerequisite:* a `WorkingTimeCalculator` object has already been initialized correctly and handed over as first argument within an XSL variable, see Methods for defining the working time
getWorkingTimeMinutes(*WorkingTimeCalculator* wtc, *int* fromYear, *int* fromMonth, *int* fromDay, *int* fromHr, *int* fromMin, *java.util.Date* to**)**	Determines the number of minutes within the defined working times, which lie between a start date (defined as integer number for year, month, day, hour and minute) and an end date of the type `java.util.Date`, and returns a number of the type `double`. *Prerequisite:* a `WorkingTimeCalculator` object has already been initialized correctly and handed over as first argument within an XSL variable, see Methods for defining the working time
getWorkingTimeMinutes(*WorkingTimeCalculator* wtc, *int* fromYear, *int* fromMonth, *int* fromDay, *int* fromHr, *int* fromMin, *int* toYear, *int* toMonth, *int* toDay, *int* toHr, *int* toMin**)**	Determines the number of minutes within the defined working times, which lie between a start and an end date (defined as integer number for year, month, day, hour and minute), and returns a number of the type `double`. *Prerequisite:* a `WorkingTimeCalculator` object has already been initialized correctly and handed over as first argument within an XSL variable, see Methods for defining the working time

5.7.2.3 Example

```
<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

Learn how the XML checking function `checkXML` of the Java class `de.softproject.xsl.XSLHelper` is used within XSL mappings.

The Java class `de.softproject.xsl.XSLHelper` is located within the X4 Server library `x4-client.jar` (under `<X4>\Server\<wildfly>\modules\system\layers\base\de\softproject\x4\client\main`).

> An XSL mapping using this help class, can't be executed in the transformation preview of the Mapping Editor.

5.7.3.1 Method Description

checkXML(*java.lang.String* URL**)**	Checks if a document, which is referenced by any valid URL as string according to `java.lang.String`, is a well-formed XML and returns a value of the type `boolean`. Valid URL protocols are: • `file:///` for local files • `xstore://` for repository documents • `context:` for variables within the context of the current X4 process instance

5.7.3.2 Example

The following XSL mapping accesses an XML document, which is stored in a process variable *`Var1`* at runtime of an associated X4 process, via the `context: URL Protocol`. Thereby, the included Java method checkXML first checks, if the document is well-formed. In the following steps the document is processed and an error handling occurs.

```
<?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 functions

In the Mapping Editor, you can use the features provided by XSLT 2.0 in your XSL mappings.
For more information, visit the World Wide Web Consortium (W3C) website.

5.9 XPath Functions

Functions provided by XPath 1.0 that can be used in XSL mappings

> Arguments within [squared brackets] are optional.

5.9.1 Node-set Functions

count(node-set)	Returns an integer number with the number of nodes in `node-set`.

id (Id)	Selects the node with the unique ID specified in `Id`. *Example:* `<xsl:value-of select="id('abc')/child::para"/>` Selects the child node `para` of the node with the unique ID *`abc`*.
last ()	Returns an integer number with the position number of the last node in the current context. *Example:* `<xsl:if test="position() = last()">` checks, if the current node is the last node.
local-name ([node-set])	Returns a string with the local part of a node name with namespace declaration. The optional argument `node-set` specifies a node-set.
name ([node-set])	Returns a string with the name of the current node. The optional argument `node-set` specifies a node-set.
namespace-uri ([node-set])	Returns a string with the namespace URI (Uniform Resource Identifier) of the current element or attribute node. The optional argument `node-set` specifies a node-set. *Example:* `<xsl:value-of select="namespace-uri()"/>` Returns the namespace URI (if existing) of the current node.
position ()	Returns an integer number with the position of the current node. The first node has the position number *`1`*.

5.9.2 Boolean Functions

Boolean XPath functions return `true/false` values and can be combined with logical operators.

boolean (Arg)	Checks if argument `Arg` is true and returns a boolean value. All arguments that are a positive number, a non-empty string or a non-empty node-set, will be interpreted as `true`.
false ()	Returns `false`.
lang (Arg)	Returns `true`, if the language specified in argument `Arg` matches at least partially the language that has been specified in an `xml:lang` attribute. *Example:* `lang("en")` returns `true`, if the current node contains the attribute `xml:lang="en"` or matches partially like `xml:lang="en-US"`.
not (Arg)	Returns the negation of a boolean expression in argument `Arg`. *Example:* `<xsl:if test="not(element > 1)">` returns `true`, if the content of `element` is less or equal to `1`.
true ()	Returns `true`.

5.9.3 Number Functions

XPath number functions return integer numbers and can be combined with logical operators.

ceiling (Arg)	Rounds up a number within the argument `Arg` to the next higher integer number and returns this integer number (e. g. `0.001` to `1`). If necessary, the argument will be converted to a number first. *Mathematically correct:* Returns the smallest integer number that is not smaller than the argument `Arg`.
floor (Arg)	Rounds down a number within the `Arg` to the next lower integer number and returns this integer number (e. g. `0.999` to `0`). If necessary, the argument will be converted to a number first. *Mathematically correct:* Returns the biggest integer number that is not bigger than the argument `Arg`.
number([Arg])	Converts the current node's content into a number and returns this number. Optionally, the argument `Arg` can specify a node-set, string, boolean value or another object to be converted. Boolean values `true` and `false` will be converted to `1` and `0`. *Example:* A string `3xy` will be converted to `3`; `xy` will be interpreted as `NaN` (Not a Number).
round (Arg)	Rounds a number within the argument `Arg` to the next positive or negative integer number and returns this number. Numbers lower than `[value].5` will be rounded down, fractions greater or equal `[value].5` will be rounded up to the next higher integer number. *Example:* `0.49` will be rounded down to `0`, `-1.5` will be rounded up to `-1`.
sum (node-set)	Creates the total sum of all node contents of the argument `node-set` and returns it as number. If necessary, the node contents will be converted to numbers first. Values that cannot be interpreted as numbers will be handled as `NaN` (Not a Number).

5.9.4 String Functions

XPath string functions handle objects and return strings or boolean values when evaluating strings.

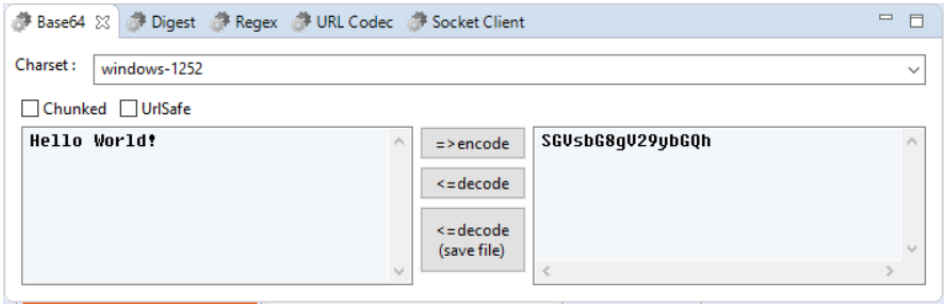
concat (Arg1, Arg2[, ...])	Joins multiple strings that are handed over as arguments `Arg1, Arg2` etc. and returns a string. If necessary, all arguments will be converted to a string first.
contains (Arg, Substring)	Checks if the argument `Substring` is contained within the argument `Arg` and returns a boolean value. If necessary, all arguments will be converted to a string first. *Example:* `<xsl:if test="contains('SoftProject', 'Project')">` returns the boolean value *`true`*.
starts-with (Arg, Substring)	Checks if the argument `Arg` starts with the argument `Substring` and returns a boolean value. If necessary, all arguments will be converted to a string first. *Example:* `<xsl:if test="starts-with('SoftProject', 'Soft')">` returns the boolean value *`true`*.

normalize-space ([Arg])	Removes leading and trailing whitespaces of the current node content, replaces multiple whitespaces, tabulators and line breaks with a single whitespace, and returns the node content as string. Optionally, an argument `Arg` can be used, which will be converted to a string, if required.
string ([Arg])	Converts the current node content into a string and returns this string. Optionally, the argument `Arg` may contain boolean values that will be converted to `true` or `false`. Decimal numbers will be simplified, if possible. Use the XSLT function `formatnumbers()` for all further number-to-string conversions. *Example:* `<xsl:value-of select="string(4.00)"/>` returns the string *4*. Check if the value is a number: `<xsl:if test="string(number(@value))!='NaN'">`.
string-length ([Arg])	Returns an integer number with the number of characters of the current node content. If necessary, all arguments will be converted to a string first. Optionally, an argument `Arg` can be used, which will be converted to a string, if required.
substring (Arg, Start, [length])	Extracts a substring of the argument `Arg` starting from the character `Start`, and returns a string. The character counting starts with `1`. Optionally, the argument `Length` can be used, which specifies the maximum length of the substring.
substring-before (Arg, Substring)	Checks if a string in argument `Arg` contains a substring within the argument `Substring` and returns a string with all characters that precede the substring. If `Substring` is not contained in `Arg`, an empty string will be returned. *Example:* `<xsl:value-of select="substring-before('2008/04/01', '/')"/>` returns *2008*.
substring-after (Arg, Substring)	Checks if a string in argument `Arg` contains a substring within the argument `Substring` and returns a string with all characters that follow the substring. If `Substring` is not contained in `Arg`, an empty string will be returned. *Example:* `<xsl:value-of select="substring-after('2008/04/01', '20')"/>` returns *08/04/01*.
translate (Arg, Substring1, Substring2)	Searches in argument `Arg` for a string within the argument `Substring1`, replaces this with the string within argument `Substring2`, and returns the modified string. If necessary, the argument `Arg` will be converted to a string first. Each matching character of `Substring1` will be replaced with the corresponding character from `Substring2`. *Example:* `<xsl:value-of select="translate('hello world', 'abcdefghijklmnopqrstuvwxyz', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')" />` returns *HELLO WORLD*.

5.10 Developer Utilities

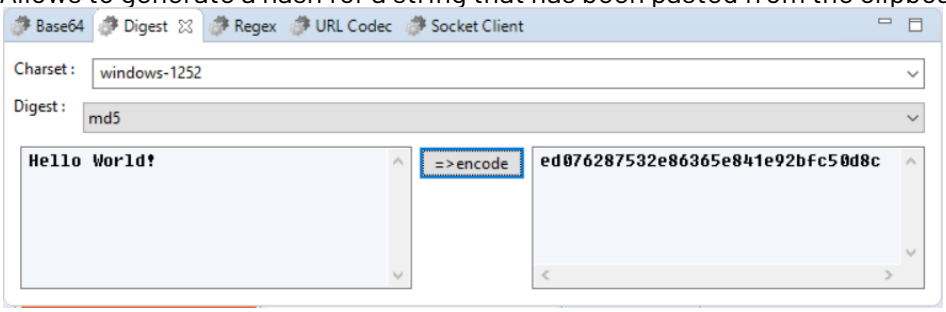
A range of developer utilities help process developers to URL-encode strings, encode/decode Base64 data, and to create hashes with a click. Within the X4 Designer, each utility is represented by a separate view, provided by the Eclipse plug-in DevUtils.

You can access the developer utilities via the menu **View**.

View	Description
Base 64	Allows any string pasted from the clipboard to be encoded or decoded as Base64  *Provided options:* • **Charset:** Encoding of the input string • **Chunked:** Output the Base64-encoded string chunked • **UrlSafe:** Only use characters that are allowed in URLs when Base64-encoding **Note:** • The button **decode (save file)** stores a decoded Base64 string as file into the local file system. • Currently, it is only possible to drag & drop files from the local file system, not from the Repository Navigator.

Digest

Allows to generate a hash for a string that has been pasted from the clipboard

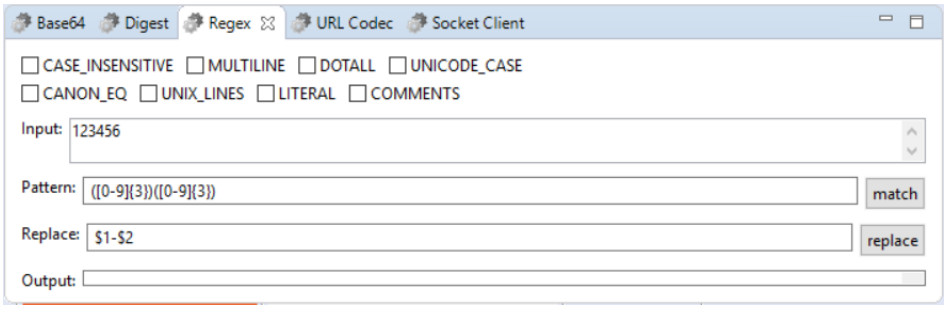


Provided options:

- **Charset:** Encoding of the input string
- **Digest:** Choose an algorithm:
  - `md5`: Generate an MD5 hash (default)
  - `sha`: Generate a hash using the Secure Hash Algorithm (SHA)
  - `sha256`: Generate an SHA hash with 256 Bit
  - `sha384`: Generate an SHA hash with 384 Bit
  - `sha512`: Generate an SHA hash with 512 Bit

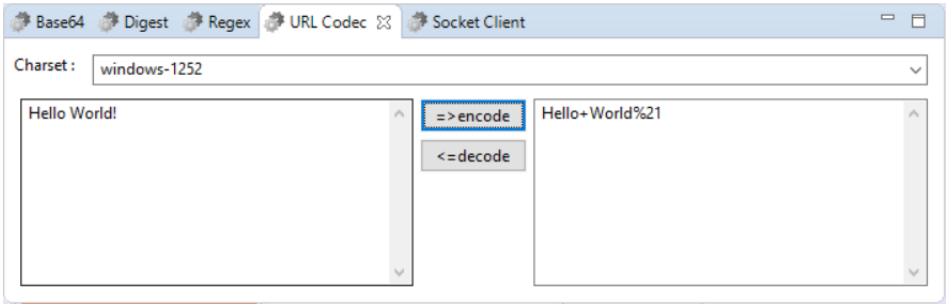
Regex

Allows to apply any regular expression (short: *Regex*) to any string that has been pasted from the clipboard. This allows you to test any expression (by clicking the **match** button), or to replace the string according to the Regex pattern (defined in **Pattern**) by clicking the **replace** button. This view also allows you to enable various Regex syntax options, such as `CASE_INSENSITIVE` or `UNICODE_CASE`.

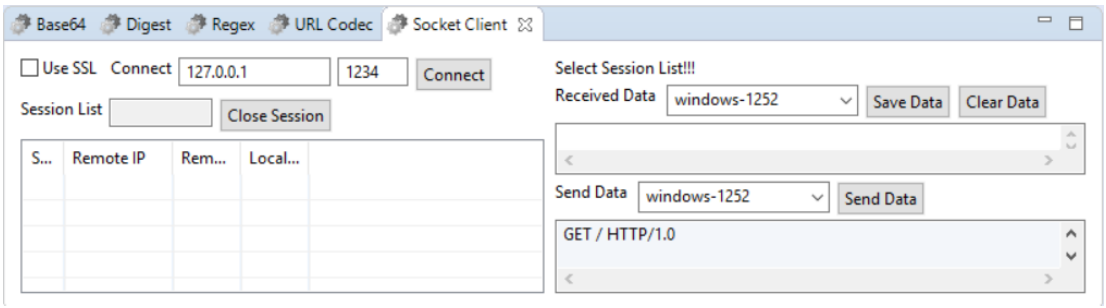


URL Codec

Allows any string pasted from the clipboard to be URL-encoded or decoded



SocketClient



5.11 URL Protocols

Learn which X4-specific URL protocols are available that enable you to access process context variables and repository files from XSL mappings during the runtime of technical processes.

5.11.1 context: URL Protocol

The `context:` URL protocol enables you to access variables from XSL mappings during the runtime of an X4 process. This variables can be defined within the process with a `Variable` component and must be placed in a process before the XSL mapping component. The document stored in a context variable can be retrieved from the XSL mapping using the `document()` function.

URL Structure

The URL must follow the pattern (slashes after the URL handler may cause problems):

```
context:VariableName
```

> ⓘ An XSL mapping using this URL protocol cannot be executed in the transformation preview oft the Mapping Editor.

Example

A sample process containing the following XSL mapping provides a variable `Var` that has been defined using a `Variable` component. This variable stores an XML document that will be retrieved from the XSL mapping using the `document()` function, and will be buffered in an XSL variable `DocVar`. Its nodes can be selected via XPath without limitation.

```
<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 Protocol

The `xstore://` URL protocol allows you to read any document within the *X4 Repository* of the current user during the process runtime. It can be used in XSL mappings and adapter parameters to reference a repository file. In XSL mappings, XML documents can be included using the `document()` function.

URL Structure

An `xstore://` URL must follow the pattern:

```
xstore://<File-path>
```

Components of an `xstore://` URL:

- `xstore://`: Protocol declaration; within the X4 BPMS valid during the runtime of the currently executed process.
- `File path`: Path of the file in the user repository folder; folders are separated by slashes (/).

> **Please note:**
>
> - To read files from other *X4 Repository* user folders, use the `x4db:/ protocol`, see x4db:/ URL Protocol.
> - An XSL mapping using this URL protocol cannot be executed in the transformation preview of the Mapping Editor.

Example

In the following example, an XSL mapping retrieves an XML document from the repository using the `document()` function and stores it in an XSL variable `externalFile`. Any XPath expression can be used to select its nodes.

```
<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 Protocol

The `x4db:/` URL protocol allows you to access during the process runtime a document in any *X4 Repository* user folder. It can be used in XSL mappings and adapter parameters to reference any repository file. In XSL mappings, XML documents can be included using the `document()` function.

URL Structure

An `x4db:/` URL must follow the pattern:

```
x4db:/<User repository>/<File path>
```

Components of an `x4db:/` URL are:

- `x4db:/`: Protocoll declaration; within the X4 BPMS valid during runtime of the currently executed process.
- `User repository`: Repository user folder (*"User collection"*), e.g. *1* or *X4P*
- `File path`: Path of the file in the user repository folder; folders are separated by slashes (`/`).

> **Please note:**
>
> - For security reasons, use the `x4db:/` URL protocol carefully, and only to read a document from another *X4 Repository* user folder. Else use `xstore://`, see xstore:// URL Protocol
> - An XSL mapping using this URL protocol cannot be executed in the transformation preview of the Mapping Editor.

Example
In the following example, an XSL mapping retrieves an XML document from the *X4 Repository* user folder `101` using the `document()` function and stores it in an XSL variable `externalFile`. Any XPath expression can be used to select its nodes.

```
<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 Protocol

The `cryptstore://` URL protocol allows you to access encrypted repository documents from within XSL mappings during process runtime, and decrypt these documents in the same step.
The document can be retrieved from an XSL mapping using the `document()` function, which allows direct access to the decrypted document.

Using a cryptstore implementation

The *X4 Server* allows to encrypt and decrypt repository elements using the `cryptstore://` URL protocol. The byte stream will be passed through a customer-specific Java class in order to be read, decrypted and outputted. This customer-specific class must be derived from `InputStream` and must have a constructor that accepts an `InputStream` as only parameter.

It is recommended to derive the class from `javax.crypto.CipherInputStream`, and to parameterize the object in the constructor. The finished class must be provided as Java archive (`.jar`) within the folder `<WildFly>\modules\system\layers\base\de\softproject\x4\client\main`.

Additionally a child element `<resource-root path="<Name des Java-Archivs>.jar"/>` must be added within the element `<resources>` of the file `module.xml` (under `<WildFly>\modules\system\layers\base\de\softproject\x4\client\main`).

Moreover, the classname must be specified as fully qualified classname in the configuration file `<X4>/X4config.xml` within an element `<cryptstoreStreamHandlerClass/>`. This element must be an immediately following sibling of the `<xstore/>` element.

URL structure

The URL must follow the pattern:

```
cryptstore://RepositoryPath/EncrypedFile
```

An XSL mapping using this URL protocol cannot be executed in the transformation preview oft the Mapping Editor.

Example

In the following example, an XSL mapping retrieves an encrypted XML document from the repository using the `document()` function and buffers it in an XSL variable `externalFile`. Its nodes can be selected via XPath without limitation.

```
<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>
```

Example for a possible implementation

```
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 Protocol

The `globalcontext:` URL protocol allows you to access existing context keys from XSL mappings during runtime of a Technical Process. Context keys can be defined with the *Global Context Manipulation* Adapter. The documents stored in the context can be included using the `document()` function.

Global context

With a globally defined context, the X4 BPMS can buffer any document from a Technical Process in a global hash table, similar to Technical Process context variables. This enables other Technical Processes to have access to this context. The documents in the global context will be stored with a key, and remain accessible during the total runtime of the *X4 Server*.

URL structure

The URL must follow the pattern:

```
globalcontext:KeyName
```

An XSL mapping using this URL protocol cannot be executed in the transformation preview oft the Mapping Editor.

Example

A sample XSL mapping retrieves a context key `X4Experience.GlobalContext.Sample` that has already been defined using the *Global Context Manipulation* Adapter component. This context entry contains an XML document that will be retrieved from the XSL mapping using the `document()` function, and will be buffered in an XSL variable `DocVar`. Its nodes can be selected via XPath without limitation.

```
<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 Technical Processes

In Technical Processes, you can log technical and business information and output log messages to the monitoring component or via Log4j to the server log, for example.

Function Adapter

Various adapters are available for logging Technical Processes.

- `Process Monitoring`: Lists all process steps and execution times of a Technical Process and outputs detailed information about a specific Technical Process.
- `Process Monitoring Management`: Updates or lists profilings that are in the X4 repository or in a project.
- `Process Runtime Management`: Stops, terminates and lists running Technical Processes.
- `Log4J Logging`: Returns data from the Technical Processes via *Log4J* to the server log. All data that can be converted into a string can serve as input data. In `AdvancedMode`, data can be output directly at various log levels in an XML document with an adapter-specific structure.

Logging action on Transitions

- In X4 Designer, you can set up logging by defining a logging action for transition lines. In the **Properties** view, you can set the `Log4j Log` property for selected transition lines to configure logging behavior.

.log file

- The `.log` file is read in at server startup.
- The file name of the `.log` file must not be changed.

- The X4 Server can write data to be logged to a `.log` file in the server directory. To activate the `.log` file, the `standalone.xml` of the X4 Server has to be adapted.

By default, the data to be logged is written to memory. The data contains the processes with process run times and process step run times. Additionally, you can configure the X4 BPMS to write the data to a `.log` file or to Elasticsearch.

The logged data can be read out with the Process Monitoring adapter and then used. However, only one source is allowed to read the data.

5.12.1 Activating .log file

- The `.log` file is read in at server startup.
- The file name of the `.log` file must not be changed.

To enable the .log file, the following changes are required in `<WildFly>\standalone\configuration\standalone.xml`.

1. In the `<subsystem xmlns="urn:jboss:domain:logging:8.0">` element, add the following section:

```
<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. In the `<logger category="X4-Profiling-Filesystem" use-parent-handlers="false">` element, add the following section:

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

3. Save standalone.xml.
4. Restart X4 Server.

5.13 Using Save Points for resumable processes

Save points allow to save the complete state of the X4 Process Engine. It stores all processed data and status information in the file system during the Technical Process instance's runtime.
If the Technical Process is resumed from a save point, the persisted data will be restored. If data, adapters, and subprocesses are invoked for the first time behind a save point, the most recent components will be used.

You can define save points by selecting a transition line in the process diagram and defining one of the following operations for the property `Save Point` in the **Properties** view.

	`Save and Continue`	Set a save point, and continue the Technical Process instance's execution immediately. In more complex Technical Processes it allows you to store intermediate results. If errors or failures occur, the Technical Process instance can be resumed from the last save point, and you can avoid to restart the whole Technical Process.
	`Save and Stop`	Set a save point, then halt the Technical Process instance's execution. This allows you to implement long-running processes, that can be "put to sleep" during the process execution until external data manipulation or maintenance work is finished. Afterwards, you can resume the Technical Process instance.
	`Delete`	Delete current save point

Restrictions

- Database transactions or cursors can not be persisted.
- For each process instance a save point can be saved. If a process contains multiple save points, the last state of the save point is used.
- Save points cannot be used in debug mode.

Resuming a process instance execution via the Monitoring interface

1. Open the Monitoring interface in your browser.
2. Select the corresponding halted process in the Instance view.
   The process will be displayed in the detail view.
3. Select the corresponding save point under **Savepoints**.

4. Click **Restart/Resume** to resume the process instance's execution.
   The process instance will be resumed based on the most recent save point, using all processed data and the persisted state of the Process Engine.

5.14 Deploying Web Services via HTTP(S)

The central web service interface allows Technical Processes to be provided as web services via HTTP(S), e.g. for ReSTful web services or SOAP web services. The individual resources of a web services can be addressed from the outside via an HTTP URL (Uniform Resource Locator).

Base URL and functionality

The base URL of the central web service interface is `http://<server>/X4/httpstarter/ReST` (e.g. `http://localhost:8080/X4/httpstarter/ReST`).

HTTP requests are processed instantly by the central web service interface and handed over as XML document to a process. The result of the process, e.g. a generated document, can be returned instantly.

Configuration

Web services are configured within the X4 Designer using the graphical *Webservice Configuration Editor*.

Return values

The web service interface returns the current status code (e.g. `201 Created`), and depending on the request, the identified resource.

Supported web service types

- **ReSTful web services**: Representational State Transfer (ReST) is an architectural style in which any resource of an information system can be addressed under its own URI, e. g. via HTTP. Thus, the target of ReSTful services within the X4 Server is to map path structures of a URI to processes in a transparent and comprehensible way.
  For ReSTful web services multiple operations or HTTP methods can be assigned for each resource and linked with a process, e.g. for accesses via `GET`, `POST` or `DELETE`. The following HTTP methods are supported:
  - `GET`
  - `PUT`
  - `POST`
  - `DELETE`
  - and any other methods with a valid XML name.

  When calling a web service, a particular XML data structure is passed to the process by default, see ReSTful Web Services: Input XML Structure. Moreover, the web service interface expects an XML document as result, which is structured according to a defined structure, and provides a number of options.

The transmitted data must be Base64 or Base16 (hexadecimal) encoded, since only bytes are transferred via HTTP. To avoid problems with the interpretation of the content within the element <body>, text or XML is directly supported here. Therefore, the text/XML document must first be converted to bytes and encoded to Base16/Base64, to have full control of the bytes that are returned as response.

> For further information regarding the ReST architecture, see the dissertation of Roy T. Fielding at http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.

- **SOAP web services**: SOAP is a W3C-standardized message protocol (http://www.w3.org/TR/soap), which enables standardized communication between distributed applications. SOAP web services can be provided in various forms, i. a. with external WSDL definition or without WSDL definition, in RPC style or in document mode. As message format for SOAP web services XML is typically used for Requests and Responses; binary data is optionally attached via MIME (MTOM).
  SOAP web services can be provided in the same way as ReSTful services via HTTP(S), see Providing SOAP Web Services.
  The service type SOAP supports the standards SOAP 1.1 and SOAP 1.2. SOAP service definitions can be combined with HTTP ReST web service definitions and with file web service definitions.
- **File web services**: The central web service interface also allows static file resources from the repository or documents generated dynamically by processes to be provided as service via HTTP GET, see Providing File Web Services.

5.14.1 Editing the Web Service Configuration graphically

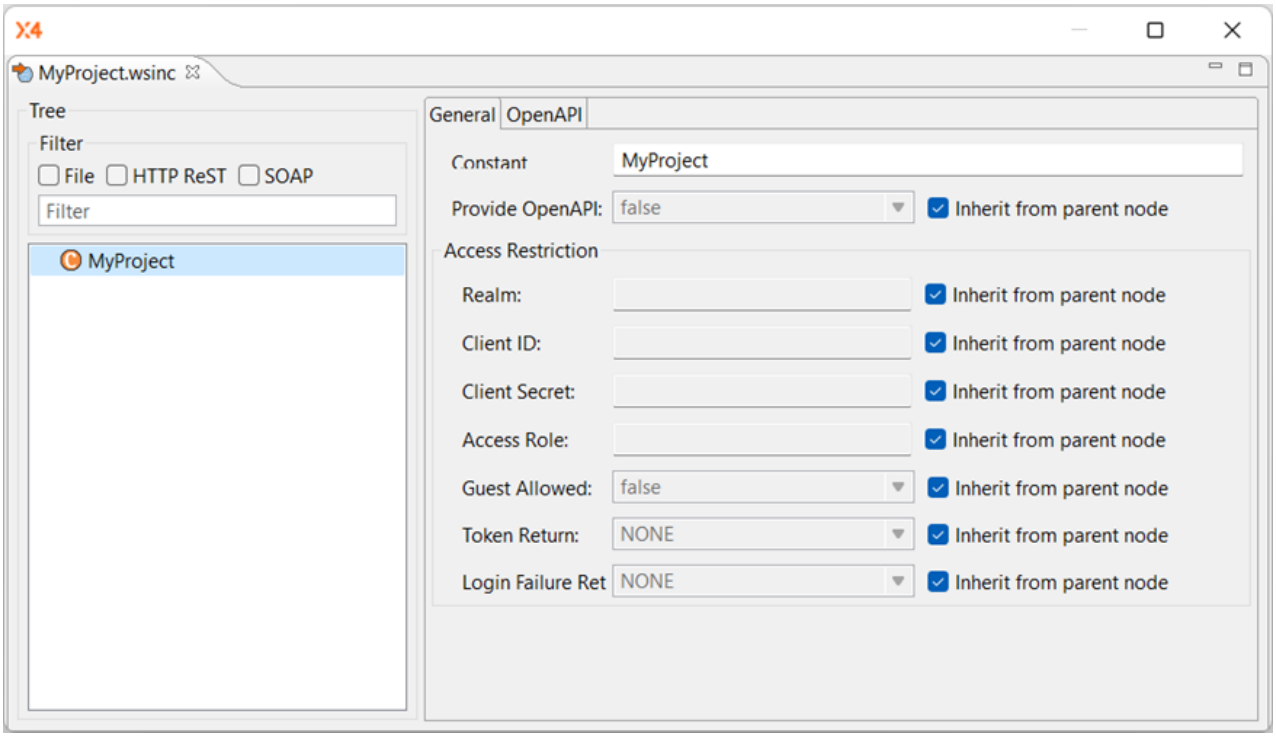
The web service configuration can be edited within the *X4 Designer* via the graphical *Webservice Configuration Editor* and updated on the *X4 Server*. Technical Processes can thus be provided as HTTP-based web services of various types (ReST, SOAP etc.) with just a few mouse clicks.
All web services provided in this way have a uniform base URL. The base URL is http://localhost:8080/X4/httpstarter/ReST by default.

5.14.1.1 Calling the Webservice Configuration Editor

To call the *Webservice Configuration Editor*, double-click the `.wsinc` file in the **Projects** view.

5.14.1.2 Interface of the Webservice Configuration Editor

The *Webservice Configuration Editor* is divided into the following sections:



Tree: URL tree structure

The **Tree** section displays the hierarchical web service URL structure, i.e. the URL parts in an expandable and collapsible tree structure. You can edit this tree structure by right-clicking on an element in the structure and choosing an option from the context menu. Moreover, you can move tree elements via drag and drop.

The **Filter** field allows you to filter the displayed tree structure by entering a node name. Structures that don't match the filter expression are hidden. You can reset the filter using the (**Reset Filter**) icon or by clearing the field. Using the **File**, **HTTP ReST**, and **SOAP** checkboxes, you can also filter by service type.

Tree elements

Constant	Constant URL part Within the **Scope** section, you can define all properties of a selected constant, or you can inherit security properties from the parent element.
Variable	Variable URL part Within the **Scope** section, you can define all properties of a selected variable, or you can inherit security properties from the parent element.

Remainder	Fully flexible URL part definition Within the **Scope** section, you can define all properties of a remainder or you can inherit security properties from the parent element.

5.14.1.3 Managing and Creating Constants, Variables or Remainders

Select the desired node (corresponds to a URL part of the web service) within the tree view and right-click to select one of the following actions from the context menu:

Add Constant	Add constant as a child node to define a constant URL part for your web service
Add Variable	Add variable as a child node to define variable parts via placeholders within the URL of your web service
Add Remainder	Add remainder as a child node to handle all remaining URL parts
Copy Service URL to Clipboard	Copy corresponding web service resource URL to clipboard
Rename	Rename the selected node
Delete	Delete the selected node

ⓘ Node names are case-sensitive!

5.14.1.4 Defining Safety Properties of the Node

In **Access Restrictions**, you can define (security) properties for the selected node if required.

ⓘ If the **Inherit from parent node** option is selected, the token configuration will be inherited from the parent element. Enable this option to set a custom value.

Realm	Security realm for the current URL part
Client ID	Keycloak client
Client Secret	Keycloak client credentials
Access Role	Keycloak role The role must be assigned to the user in Keycloak to gain access to the web service.
Guest Allowed	If a security realm has been defined in the property `Realm`, it can be useful to allow guest access by selecting the value *`true`*.

Token Return	Select the returned login token **Possible values:** • *NONE* (default) • *CUSTOM_HEADER*
Login Failure Return	Combined with the Login module for *X4 Web Apps*, you can use **Login Failure Return** to control if the reason should also be returned after a failed login attempt. **Possible values:** • *NONE* (default): Do not return information about the reason of failure (logged in the server log) • *HEADER*: Return the technical or business reason in the HTTP header `X-X4-Login-Failed`.

5.14.2 Providing ReSTful Web Services

For each URL part of your ReSTful web service, you can assign an X4 handler Technical Process to each HTTP method, e.g. a `GET` entry for the `GET` method.

Available HTTP methods

You can define the following HTTP methods once within an operation block:

- `GET`
- `PUT`
- `POST`
- `DELETE`
- as well as any other methods with a valid XML name

5.14.2.1 Deploying Processes as ReSTful HTTP Web Service

To provide a Technical Process as operation of a ReSTful web service within the Webservice Configuration Editor, proceed as follows:

1. If needed, create a new node (corresponds to a URL part of the web service) within the tree structure, see Editing the Web Service Configuration graphically.
2. Choose **Add HTTP ReST Service** to create a new *HTTP ReST* service for the desired node (with the type `Constant`,`Variable` or `Remainder`) and adjust the (security) properties, if required.

3. Choose **Add HTTP ReST Operation** in the context menu to create a new operation for the HTTP ReST service or select an entry in the **Operations** table, which shall be overwritten.

   > ⓘ You can assign an operation `GET`, `POST`, `PUT`, `DELETE` once for each service or enter any operation name; it will be distinguished between upper case and lower case.

4. Drag and drop a process from the repository to the desired operation in order to assign the process path.
5. Define an input mode and an output mode for each operation to specify which XML structure is expected and created.
6. If required, enable the use of placeholders in order to enable parameter handling for the web service operation, and to hand over HTTP request information to the process context as placeholders.

   > ⓘ If option `Enabled` is selected, the following placeholders with HTTP request information are are available within the X4 process context, e.g. within XSL mappings:
   > - `REQUEST_PARAMETER_<`*`Name`*`>`: Desired parameter (case-sensitive!)
   > - `REQUEST_HEADER_<`*`NAME`*`>`: Desired HTTP header (only uppercase!)
   > - `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. Define security options such as `Realm` or `Guest Allowed` for the operation.

Example: ReSTful web service configuration (excerpt from XML storage format)

```
<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 Web Services: Input XML structure

When a web service is called, a document is transmitted to the corresponding Technical Process. If you have defined `HTTP Content` as the input mode in the web service configuration, this is only the content of the HTTP entity. `Media Type` and `Charset` from the `Content-Type` header are transmitted to the input document.
If `HTTP-MetaXml` has been defined as the input mode, an XML document with HTTP header information, among other things, is passed to the Technical Process during the web service call, which is structured according to the following pattern:

```
<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>
```

The `<Request>` root element contains the following elements:

- `<Path>`: Contains the calling URL and the individual broken down URL parts.
- `<Headers>`: Contains each HTTP headers with name and value in an `<Header>` element.
- `<Query>`: Contains the elements `<QueryString>` and, if necessary, for each parameter an element `<Parameter>`.

- `<Security>`: Contains the assigned JAAS subjects (simple principals or principal groups) if it is not a guest access.
- `<Body>`: Contains the `encoding` type *base64* in attribute encoding and the correspondingly encoded user data as element content.

5.14.2.3 ReSTful Web Services: Expected Output XML

If the output handler `HTTP-MetaXml` has been defined for the respective URI operation within the web service configuration file, the webservice interface expects an XML document with the following structure as result from the process:

```
<Response>
   <Status code="HTTP-Status-Code"/>
   <Headers>
      <Header>
         <Name>HTTP Header name</Name>
         <Value>Header value</Value>
      </Header>
      <!-- Or for date values: -->
      <Header>
         <Name>HTTP Header name</Name>
         <Date timezone="Timezone">Header date value</Date>
      </Header>
   </Headers>
   <Content encoding="Data encoding">Submitted data</Content>
</Response>
```

The root element `<Response>` contains the following elements:

- `<Status>`: Contains in its attribute `code` a three-digit number with the HTTP status code (e. g. *202*).
- `<Headers>`: Contains any number of `<Header>` elements with HTTP header information. Each `<Header>` element contains an HTTP header, which is defined via the child elements `<Name>` and `<Value>` or `<Date>` (for date values in the format `yyyy-MM-dd HH:mm:ss` according to *SimpleDateFormat*). For the element `<Date>`, an optional attribute `timezone` can specify a timezone.
- `<Content>` (optional): Contains within the attribute `encoding` the encoding type (*base64* or *base16* respectively *hex*) and as element content the correspondingly encoded data and/or the stream elements.

```
<Content encoding="base64">
    ... Base64...
    <Stream url="http://remote.host/Some/Document?id=foo" transcoding="base64-
us-ascii"/>
    ... mehr Base64...
</Content>
```

5.14.3 Providing SOAP Web Services

Each SOAP web service is defined via a `<Services/>` node within the webservice configuration. Its attributes and child elements vary depending on the web service type.

The following SOAP web service types are available within the `<Wsdl>` child element's attribute `mode`:

- `RPC`: The X4 Server generates a WSDL definition from the configuration data according to the RPC style, see Providing SOAP Web Services in RPC-Style
- `DOCUMENT`: The X4 Server generates a WSDL definition from the configuration data according to the Document style, see Providing SOAP Web Services in Document Mode
- `EXTERNAL`: Use external WSDL definition for the SOAP web service, see Providing SOAP Web Services with External WSDL Definition
- `NONE`: No WSDL definition will be used/provided, see Providing SOAP Web Services without WSDL Definition
- `PROCESS`: Select a process which creates the WSDL and returns it to the caller as HTTP content. The process result is thereby returned in its original form.

Schematic structure of a SOAP web service configuration (XML storage format)

```
<Service type="Soap">
  <SoapVersion>
    <SOAP11/> <!-- for SOAP 1.1 support -->
    <SOAP12/> <!-- for SOAP 1.2 support -->
  </SoapVersion>
  <Wsdl mode="WSDL mode"/>
  <Namespace prefix="" namespace="" schemaLocation="URL to an XSD document"/>
  <Operation name="Operation name" process="Path to the X4 process as x4db:/ URL">
    <InputHandler type="Input mode"/>
    <OutputHandler type="Output mode"/>
    <Input>
      <Part name="Part name" type="Type (optional)" element="Element name
(optional)" />
    </Input>
    <Output>
      <Part name="Part name" type="Type (optional)" element="Element name
(optional)" />
    </Output>
  </Operation>
</Service>
```

Explanations:

- The node`<Services type="Soap">` defines an HTTP web service of the type SOAP.
- The node `<Wsdl>` defines with its attribute `mode` the desired WSDL mode (e.g. `EXTERNAL`, see above).
- The nodes `<SOAP11/>` and `<SOAP12/>` allow to define the SOAP versions used by the web service nutzt (either one or both parallel).

- In any number of `<Namespace>` nodes a namespace for the definition of web services can be used.
- In any number of `<Operation>` nodes an operation for the web service can be defined (see above).

The following namespaces are defined by the system:

- `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 for SOAP Services

- **`GET` Requests**: In WSDL mode `NONE`, a `GET` request is responded with the HTTP status `405` (met hod not allowed). In WSDL mode `EXTERNAL`, the response to the configured WSDL URL is an HTTP redirect. In case of the WSDL modes `RPC` and `DOCUMENT`, the X4 Server generates a WSDL from the service configuration as response.
- `POST` Requests: In case of a `POST` request the complete request body is loaded to the memory, in order to initiate the routing on the operation.
  a. First, it is looked for the SOAP document.
     - If it is not a multipart request, it is assumed that the request body contains the SOAP document.
     - If it is a multipart request (MTOM), it is looked for the MTOM package's start part.
  b. The document found is read as XML and the SOAP version from the root element's namespace is determined. If it is not an accepted SOAP versions, it will be replied with an appropriate SOAP fault.
  c. The `Namespace` and the `Localname` of the SOAP body's first child element is figured out from the XML document, and used to determine the operation within the service.

     > - In case of the WSDL modes `NONE`, `RPC` and `EXTERNAL`, it is compared with the configured `Namespace` and `Localname` of the operation.
     > - For the WSDL mode `DOCUMENT` the configuration of the first Input part is used. If it is an *Element* part (i.e. the attribute `element` contains a value) the *Qualifiedname* of the configured element is used. For *Type* parts (i.e. the attribute `type` contains a value) the part's name (attribute `name`) is used.

  d. Then, the request ist prepared according to the input mode and passed to the configured process for the operation.

     > SOAP requests with an empty `Body` element can't be processed.

5.14.3.1 Providing SOAP Web Services in RPC-Style

Learn how to provide a Technical Process as SOAP web service in RPC style with the X4 BPMS, which also provides a WSDL definition generated automatically from the service configuration.

Prerequisites:

- Basic knowledge of the structure of SOAP web services in RPC style
- A Technical Process per web service operation, which assumes the processing of the incoming request. The Technical Process is referenced via its `x4db:/ URL`, thereby.
- A URL part, under which the SOAP web service shall be accessible via HTTP, e.g. `http://localhost:8080/X4/httpstarter/ReST/MyRPCWebservice`. Normally, you have already defined this URL part as constant within the webservice configuration.

In order to deploy a Technical Process as SOAP web service in RPC style, proceed as follows

1. Create a new *SOAP* service entry under the desired URL resource (defined e.g. by means of a constant URL part) via the context menu **Add SOAP Service**.
2. Specify the SOAP version (either *SOAP 1.1*, *SOAP 1.2* or support for both standards).
3. Select the WSDL mode in **WSDL Mode** and the target prefix in **Service Prefix**.
4. Define the following configuration for each operation:
   - Specify a valid name for the operation.
     SOAP web services in RPC style require explicit *Localnames* for the operations.
   - Link the Technical Process which handles the web service request via the X4 Server using its `x4db:/ URL`.
     Within the *Webservice Configuration Editor* this is possible via drag & drop from the *Repository Navigator*.
   - Define the input and output handler for the SOAP web service.
     Example: Define the *SOAP-Content* to use the first element within the `soap:Body` as input and to create a `soap:Body` and a `soap:Envelope` element for the output.
   - Specify the expected input XML data structure within the element `<Input>`, optionally with XML schema definition.
   - Specify the output data structure to be generated within the element `<Output>`.
5. Save and apply the web service configuration with **Reload on Server**.
   The SOAP web service in RPC style will be active instantly.

Generated WSDL Definitions in RPC

The automatically generated corresponding WSDL definition uses the `RPC` style and is output in case of an `HTTP GET` call of the SOAP web service URI.

For the WSDL mode `RPC`, the X4 Server generates a WSDL definition from the service configuration. The WSDL definition is created as follows:

- For each namespace element with an attribute `schemaLocation` an `xsd:schema` with `xsd:include` for the `schemaLocation` will be generated under `wsdl:types`.

- For each operation a `wsdl:message` will be generated whose name corresponds to the *Localname* of the configured operation. For each `Input` part a corresponding `wsdl:part` will be generated within this message.
- For each operation a `wsdl:message` will be generated whose name corresponds to the *Localname* of the configured operation, supplemented by `Response`. For each `Output` part a corresponding `wsdl:part` will be generated within this message.
- For each operation a `wsdl:operation` and thus a `wsdl:portTyp` will be generated. The name of the `wsdl:operation` is the *Localname* of the configured operation. The above generated messages are used as input/output message.
- Depending on the SOAP version, corresponding `wsdl:bindings` are generated (exclusively`literal` bindings). An empty SOAP action is defined at all times. In case of SOAP web services with RPC style, for each operation a `targetNamespace` is declared, if the operation name is a *Qualifiedname* (i.e. a name with colon).

Example: Configuration of a SOAP web service in RPC mode

```
<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 Providing SOAP Web Services in Document Mode

Learn how to provide SOAP web services in Document mode with the X4 BPMS.

Prerequisites:

- Basic knowledge of the structure of SOAP web services in Document style, in which the Request/Response Body is described completely via an XML Schema (XSD).
- A Technical Process per web service operation, which assumes the processing of the incoming request. The Technical Process is referenced via its `x4db:/ URL`, thereby.
- A URL part, under which the SOAP web service shall be accessible via HTTP, e.g. `http://localhost:8080/X4/httpstarter/ReST/MyDocumentWebservice`. Normally, you have already defined this URL part as constant within the webservices configuration, see Editing the Web Service Configuration graphically.

In order to provide a Technical Process as SOAP web service in Document mode, proceed as follows:

1. Create a new *SOAP* service entry under the desired URL resource (defined e.g. by means of a constant URL part) via the context menu **Add SOAP Service**.
2. Specify the SOAP version (either *SOAP 1.1*, *SOAP 1.2* or support for both standards).
3. Select the WSDL mode in **WSDL Mode** and the target prefix in **Service Prefix**.
4. Define the following configuration for each operation:
   - Specify a valid name for the operation.
     SOAP web services in Document mode require explicit *Localnames* for the operations.
   - Specify the namespace declaration for the data within the Request/Response body.
   - Link the Technical Process which handles the web service request using its `x4db:/ URL`.
     Within the *Webservice Configuration Editor* this is possible by drag & drop from the Repository Navigator.
   - Define the input handler and output handler for the SOAP web service.
     Example: Define the *SOAP-Content* to use the first element within the `soap:Body` as input and to create a `soap:Body` and a `soap:Envelope` element for the output.
   - Specify the expected input XML data structure including the namespace within the element `<Input>`.
   - Specify the output data structure to be generated including the namespace within the element `<Output>`.
5. Save and apply the web service configuration with **Reload on Server**.
   The SOAP web service in Document mode will be active instantly.

Generated WSDL Definitions in Document Mode

The automatically generated corresponding WSDL definition uses the `DOCUMENT` mode and is output in case of an `HTTP GET` call of the SOAP web service URL.

For the WSDL mode `DOCUMENT`, the *X4 Server* generates a WSDL definition from the service configuration. The WSDL definition is created as follows:

- For each namespace element with an attribute `schemaLocation` an `xsd:schema` with `xsd:include` for the `schemaLocation` will be generated under `wsdl:types`.
- For each operation a `wsdl:message` will be generated whose name corresponds to the *Localname* of the configured operation. For each `Input` part a corresponding `wsdl:part` will be generated within this message.
- For each operation a `wsdl:message` will be generated whose name corresponds to the *Localname* of the configured operation, supplemented by `Response`. For each `Output` part a corresponding `wsdl:part` will be generated within this message.
- For each operation a `wsdl:operation` and thus a `wsdl:portTyp` will be generated. The name of the `wsdl:operation` is the *Localname* of the configured operation. The above generated messages are used as input/output message.
- Depending on the SOAP version, corresponding `wsdl:bindings` are generated (exclusively`literal` bindings). An empty SOAP action is defined at all times.

Example: Configuration of a SOAP web service in Document mode

```
<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 Providing SOAP Web Services with External WSDL Definition

Learn how to provide SOAP web services (e.g. in RPC style) based on a consisting WSDL definition with the X4 BPMS.

Prerequisites:

- Basic knowledge of the structure of SOAP webservices.
- The WSDL definition must be valid and accessible for the consumer via a valid URL, e.g. via HTTP.
- A Technical Process per web service operation, which assumes the processing of the incoming request. The Technical Process is referenced via its `x4db:/ URL`, thereby.
- A URL part, under which the SOAP web service shall be accessible via HTTP, e.g. `http://localhost:8080/X4/httpstarter/ReST/ MyExternalWSDLWebservice`. Typically, you have already defined this URL part as constant within the webservice con figuration, see Editing the Web Service Configuration graphically.

In order to provide a Technical Process as SOAP web service with external WSDL definition, proceed as follows:

1. Create a new *SOAP* service entry under the desired URL resource (defined e.g. by means of a constant URL part) via the context menu **Add SOAP Service** and adjust the (security) properties, if required.
2. Specify the SOAP version (either *SOAP 1.1*, *SOAP 1.2* or support for both standards).
3. Select the WSDL mode `EXTERNAL` and enter a valid URL to the WSDL definition in **WSDL URL**.

You can can also provide a self-created WSDL definition and an XSD from the repository. Provide the WSDL definition file as file service via HTTP for this purpose, see Providing File Web Services.

Example (Excerpt from XML storage format)

```
<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>
```

You can then reference the file web service URL within the SOAP web service definition e.g. as follows:

Example (Excerpt from XML storage format)

```
<Wsdl mode="EXTERNAL" url="http://localhost:8080/X4/httpstarter/Rest/
soapService/externalWsdl"/>
```

4. You can click on **Load Configuration from WSDL** within the *X4 Webservice Configuration Editor* to load the WSDL definition with the corresponding web service information (SOAP version, service prefix, linked XSD schemas and available web service operations).
   For each web service operation defined in WSDL, a correspondingly named child node will be generated within the tree structure of the *X4 Webservice Configuration Editor*.
5. Adjust the configuration for each web service operation within the corresponding child node:
   - Link the Technical Process which handles the web service request via the X4 Server using its `x4db:/ URL`.
     Within the *Webservice Configuration Editor* this is possible via drag & drop from the repository.
   - Define the input and output handler for the SOAP web service.
   - Adjust the safety realms (*Guest Allowed*, *Token Return* etc.), if required.
6. Save and apply the web service configuration via **Reload on Server**.
   The SOAP web service will be active instantly and can be called by means of its service URL via HTTP POST.

In case of the service URL's call via HTTP `GET`, the URL specified within **WSDL URL** (attribute `url`) will be forwarded (HTTP redirect).

Example: Configuration of the web service with external 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 Providing SOAP Web Services without WSDL Definition

Learn how to provide SOAP web services (e.g. in RPC style) without WSDL definition via HTTP POST with the X4 BPMS.

Prerequisites:

- Basic knowledge of the structure of SOAP web services
- A Technical Process per web service operation, which assumes the processing of the incoming request. The Technical Process is referenced via its `x4db:/ URL`, thereby.
- A URL part, under which the SOAP web service shall be accessible via HTTP, e.g. `http://localhost:8080/X4/httpstarter/ReST/MyWebservice`. Typically, you have already defined this URL part as constant within the webservice configuration, see Editing the Web Service Configuration graphically.

In order to provide a Technical Process as SOAP web service without WSDL definition, proceed as follows:

1. Create a new *SOAP* service entry under the desired URL resource (defined e.g. by means of a constant URL part) via the context menu **Add SOAP Service**.

2. Specify the SOAP version (either *SOAP 1.1*, *SOAP 1.2* or support for both standards).
3. Select the WSDL mode `NONE`.
4. Define the following configuration for each operation:
   - Specify the *Qualified Name* of the first element within the SOAP request's body as name for the operation.
   - Link the Technical Process which handles the web service request via the X4 Server using its `x4db:/ URL`.
     Within the *Webservice Configuration Editor* this is possible by drag & drop from the repository.
   - Define the input and output handler for the SOAP web service.
5. Save and apply the web service configuration via **Reload on Server**.
   The SOAP web service will be active instantly and can be called by means of its service URL exclusively via HTTP POST. A call of the service URL via HTTP GET doesn't return any WSDL definition.

Example: Configuration of the web service without 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 Web Services: Input and Output XML

For each SOAP web service an input/output handler can be defined within the web service configuration (nodes `<InputHandler/>` and`<OutputHandler/>`). They control which XML data structure is expected, respectively created during the transfer to webservices.

Mode SOAP-Envelope: Input Handler

If the mode *SOAP-Envelope* has been defined as input handler (XML storage format: `<InputHandler type="SOAP-Envelope"/>`), the complete XML of the SOAP request will be passed, including header information and content of the body.

```
 <?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>
```

Mode SOAP-Envelope-Validated: Input Handler

If the mode *SOAP-Envelope-Validated* has been defined as input handler, the complete XML document of the SOAP request will be handed over, including header information and content of the body. Moreover, the input is validated against the WSDL definition and its XSD schemas.

If validation errors occur, the result document will contain a SOAP Fault element (e.g. within an element `<SOAP-ENV:Fault>`) with the error messages.

> This mode requires the service configuration field **WSDL URL** to contain a valid WSDL URL, and the WSDL has been made available via **Load Configuration from WSDL**. See also Providing SOAP Web Services with External WSDL Definition.

Mode SOAP-Envelope: Output Handler

Similar to the input handler, if the mode *SOAP-Envelope* has been defined as output handler (XML storage format: `<OutputHandler type="SOAP-Envelope"/>`), the Technical Process is responsible for the complete SOAP document's creation and thus for the creation of the `SOAP-Envelope`, `SOAP-Header` and `SOAP-Body`.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
     <SomeHeader />
  </soap:Header>
  <soap:Body>
    <SomeData />
  </soap:Body>
</soap:Envelope/>
```

Mode SOAP-Content: Input Handler

If the mode `SOAP-Content` has been defined as input handler (XML storage format: `<InputHandler type="SOAP-Envelope"/>`), the first node of the Body will be passed within the SOAP request.

EXample: 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/>
```

Example: Process input

```
<?xml version="1.0" encoding="UTF-8"?>
<GetEntry>
   <id>101</id>
</GetEntry>
```

Mode SOAP-Content-Validated: Input Handler

If the mode `SOAP-Content-Validated` has been defined as input handler, the first node of the Body will be passed within the SOAP request. Moreover, the input is validated against the WSDL definition and its XSD schemas.

If validation errors occur, the result document will contain a SOAP Fault element (e.g. within an element `<SOAP-ENV:Fault>`) with the error messages.

> This mode requires the service configuration field **WSDL URL** to contain a valid WSDL URL, and the WSDL has been made available via **Load Configuration from WSDL**. See also Providing SOAP Web Services with External WSDL Definition.

Mode SOAP-Content: Output Handler

Similar to the input handler, if the mode `SOAP-Content` has been defined as output handler (XML storage format: `<OutputHandler type="SOAP-Content"/>`), the result of the Technical Process is taken as content of the SOAP body for the SOAP response.

```
<?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>
```

Transformation of the Process Result into the Response Body

If the process result is `<SomeData/>` and it is a SOAP 1.1 request, the following response will be sent:

```
<SOAP11:Envelope xmlns:SOAP11="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP11:Body>
    <SomeData/>
  </SOAP11:Body>
</SOAP11:Envelope/>
```

If the process result is `<SomeData/>` and it is a SOAP 1.2 request, the following response will be sent:

```
<SOAP12:Envelope xmlns:SOAP12="http://www.w3.org/2003/05/soap-envelope">
  <SOAP12:Body>
    <SomeData/>
  </SOAP12:Body>
</SOAP12:Envelope/>
```

5.14.4 Providing File Web Services

Besides the possibility of providing X4 processes as ReSTful services via HTTP(S), the X4 Webservices interface also allows to provide static file resources from the repository as service via HTTP GET.

Prerequisites:

- A file within the repository, which is to be provided as file service via HTTP. The file is referenced via its `x4db:/ URL`, thereby.
- A URL part, under which the file shall be accessible via HTTP, e.g. `http://localhost:8080/X4/httpstarter/ReST/MyFile`. Normally, you have already defined this URL part (as constant, variable or remainder) within the webservice configuration, see Editing the Web Service Configuration graphically.

5.14.4.1 Providing File Resources via HTTP

1. Create a new *File* service entry under the desired URL resource (defined e.g. by means of a constant or variable URL part) via the context menu **Add File Service**.
2. Specify the `x4db:/` path to the desired file, whose content shall be provided unalteredly as file resource via HTTP, within the repository in **Path**.

   Within the *X4 Webservice Configuration Editor, y*ou can assign a file from the repository by drag & drop.
3. Adjust the security realms (*Guest Allowed, Token Return* etc.), if required.
4. Save and apply the web service configuration with **Reload on Server**.
   The changes will be active instantly and the file web service will be provided.

5.14.4.2 Using File Web Services with Variables

The path specification to the repository file within the attribute `path` can be built dynamically by defining the file service in a variable block, for example for changing file names or IDs within a variable URL part. The assignment is done based on the variable name, e.g. `<myVar>`.

Example: Definition of a FILE service as constant (excerpt from the restconfig.xml)

```
<Variable name="myVar">
    <Service type="File" path="x4db:/1/Project/Folder/&lt;myVar&gt;"/>
</Variable>
```

Explanations:

- The element `<Variable/>` defines within its attribute `name` the name of a variable, e.g. `myVar`.
- In attribute `path` this variable is referenced as placeholder at any place within the `x4db:/` URL; multiple mentions of the placeholder are also possible.
- The placeholder always begins with an opening angle bracket < (`<` in XML storage format), followed by the variable name and a closing angle bracket >.

Example: Variable placeholder at any position (excerpt from 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>
```

Example: Combined variables (excerpt from XML storage format)

```
<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>
```

5.14.4.3 Using File Web Services with Remainder

If you want to define a file web service with any URL parts, you can use the *Remainder* construct. As a more tolerant variant of a variable, it allows to intercept URL parts of any depth. Similar to variables, you define a remainder with a name and reference it within the attribute `path`.

Example: Definition of a FILE service with remainder (excerpt from XML storage format)

```
<Remainder name="myRemainderVar">
    <Service type="File"
     path="x4db:/1/Project/Folder/&lt;myRemainderVar>"/>
</Remainder>
```

Explanations:

- The element `<Remainder/>` defines within the attribute `name` the name of a remainder variable, e.g. `myRemainderVar`.
- In attribute `path` this variable is referenced as placeholder at any place within the `x4db:/` URL; multiple mentions of the placeholder are also possible.
- The placeholder always begins with an opening angle bracket < (or `<` in XML storage format), followed by the variable name and a closing angle bracket >.

Example: Further examples for remainders (excerpt from XML storage format)

```
<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>
```

5.14.4.4 Caching of Provided File Resources

Each file resource provided by means of a file web service via HTTP GET has a validity of 10 hours. Thus, the client (e.g. a Web browser) re-invokes the resource only after the expiration of this period.

To load the current resource for development or test purposes, set back the cache within the client manually or deactivate it.

5.15 Providing Web Services via Open API

The central web service interface allows Technical Processes to be provided as web services via Open API.

Base URL and functionality

The base URL of the central web service interface is `http://<server>/X4/httpstarter/ReST` (e.g. `http://localhost:8080/X4/httpstarter/ReST`).

Configuration

Web services are configured within the X4 Designer using the graphical *Webservice Configuration Editor*. For more information, see Editing the Web Service Configuration graphically.

5.15.1 Editing the Web Service Configuration graphically

The web service configuration can be edited within the *X4 Designer* via the graphical *Webservice Configuration Editor* and updated on the *X4 Server*. Technical Processes can thus be provided as OpenAPI-based web services with just a few mouse clicks.

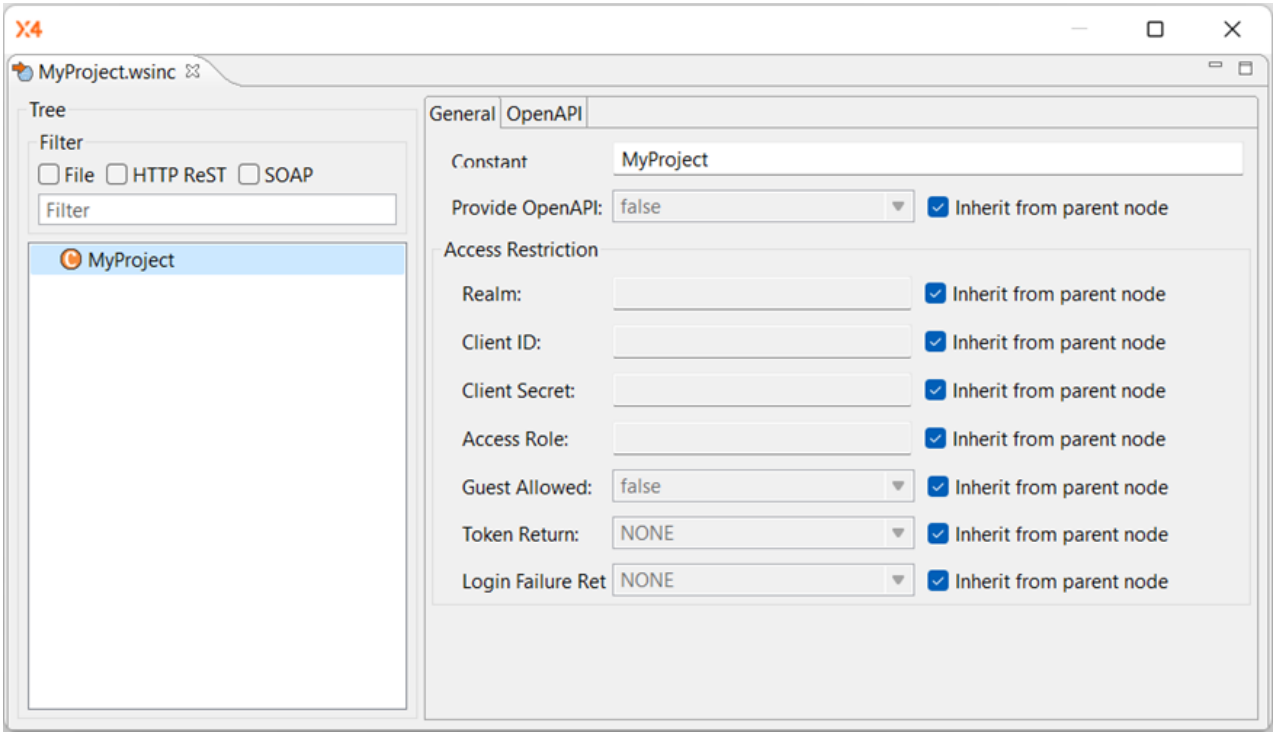
All web services provided in this way have a uniform base URL. By default, the base URL is http://localhost:8080/X4/httpstarter/ReST.

5.15.1.1 Calling the Webservice Configuration Editor

To call the *Webservice Configuration Editor*, double-click the `.wsinc` file in the **Projects** view.

5.15.1.2 Interface of the Webservice Configuration Editor

The *Webservice Configuration Editor* is divided into the following sections:



Tree: URL tree structure

The **Tree** section displays the hierarchical web service URL structure, i.e. the URL parts in an expandable and collapsible tree structure. You can edit this tree structure by right-clicking on an element in the structure and choosing an option from the context menu. Moreover, you can move tree elements via drag and drop.

The **Filter** field allows you to filter the displayed tree structure by entering a node name. Structures that don't match the filter expression are hidden. You can reset the filter using the

(**Reset Filter**) icon or by clearing the field. Using the **File**, **HTTP ReST**, and **SOAP** checkboxes, you can also filter by service type.

Tree elements

Constant	Constant URL part Within the **Scope** section, you can define all properties of a selected constant, or you can inherit security properties from the parent element.
Variable	Variable URL part Within the **Scope** section, you can define all properties of a selected variable, or you can inherit security properties from the parent element.
Remainder	Fully flexible URL part definition Within the **Scope** section, you can define all properties of a remainder or you can inherit security properties from the parent element.

5.15.1.3 Managing and Creating Constants, Variables or Remainders

Select the desired node (corresponds to a URL part of the web service) within the tree view and right-click to select one of the following actions from the context menu:

Add Constant	Add constant as a child node to define a constant URL part for your web service
Add Variable	Add variable as a child node to define variable parts via placeholders within the URL of your web service
Add Remainder	Add remainder as a child node to handle all remaining URL parts
Copy Service URL to Clipboard	Copy corresponding web service resource URL to clipboard
Rename	Rename the selected node
Delete	Delete the selected node

ⓘ Node names are case-sensitive!

5.15.1.4 Defining Safety Properties of the Node

In **Access Restrictions**, you can define (security) properties for the selected node if required.

If the **Inherit from parent node** option is selected, the token configuration will be inherited from the parent element. Enable this option to set a custom value.

Realm	Security realm for the current URL part
Client ID	Keycloak client
Client Secret	Keycloak client credentials
Access Role	Keycloak role The role must be assigned to the user in Keycloak to gain access to the web service.
Guest Allowed	If a security realm has been defined in the property `Realm`, it can be useful to allow guest access by selecting the value *`true`*.
Token Return	Select the returned login token **Possible values:** • *`NONE`* (default) • *`CUSTOM_HEADER`*
Login Failure Return	Combined with the Login module for *X4 Web Apps*, you can use **Login Failure Return** to control if the reason should also be returned after a failed login attempt. **Possible values:** • *`NONE`* (default): Do not return information about the reason of failure (logged in the server log) • *`HEADER`* : Return the technical or business reason in the HTTP header `X-X4-Login-Failed`.

5.15.2 Providing ReSTful Web Services

For each URL part of your ReSTful web service, you can assign an X4 handler Technical Process to each HTTP method, e.g. a `GET` entry for the `GET` method.

Available HTTP methods

You can define the following HTTP methods once within an operation block:

- `GET`
- `PUT`
- `POST`
- `DELETE`
- as well as any other methods with a valid XML name

5.15.3 Deploying a Technical Process ass OpenAPI ReST Service

Prerequisite:
You provide of an ESB project with at least one Technical Process.

In order to provide a Technical Process as an operation of an OpenAPI ReST Web Service, proceed as follows in the *Webservice Configuration Editor*:

1. Enter the desired data in the **OpenAPI** tab.

   > ⚠ **Name**, **Title**, and **Version** are mandatory field which require a value. If a method description is to be displayed in the API, please enter a description in the **Description** field.

2. If required, create a new node in the tree structure (corresponds to an URL part of the web service). For more information, see Editing the Web Service Configuration graphically.
3. Use **Add HTTP ReST Service** to create a new service of the *HTTP ReST* type for the desired node (of the `Constant`, `Variable`, or `Remainder` type). Adjust the (security) properties, if needed.

   > ⚠ Before creating an HTTP ReST service, you must set the **Provide OpenAPI** value on the **General** tab to **true** and uncheck the **Inherit from parent node** checkbox. By doing so, you specify that you use OpenAPI in the HTTP ReST service.

4. Use the **Add HTTP ReST Operation** context menu to create a new operation for the HTTP ReST Service or choose an existing entry from the **Operations** that is to be overwritten.

   > ⓘ You can assign a `GET`, `POST`, `PUT`, or `DELETE` operation once for each service or enter any operation name (the entry is case-sensitive).

5. Drag a process from the **Projects** view to the desired operation via drag-and-drop in order to assign the process path.
6. Use the **Output Mode** drop-down to set the `HTTP-Content` output mode in order to display the process response.
7. If the ReST service requires query parameters, you can create them. To do so, click the **Add** button in the **Query Parameter** group box in the **OpenAPI** tab. When adding new parameters, you must specify a name for the parameter and additional options, such as **Type**, **Array**, and **Required**.
8. You can use the **Request** and **Response** tabs to create a custom request and a custom response for each method.

5.15.3.1 Working with a Data Model

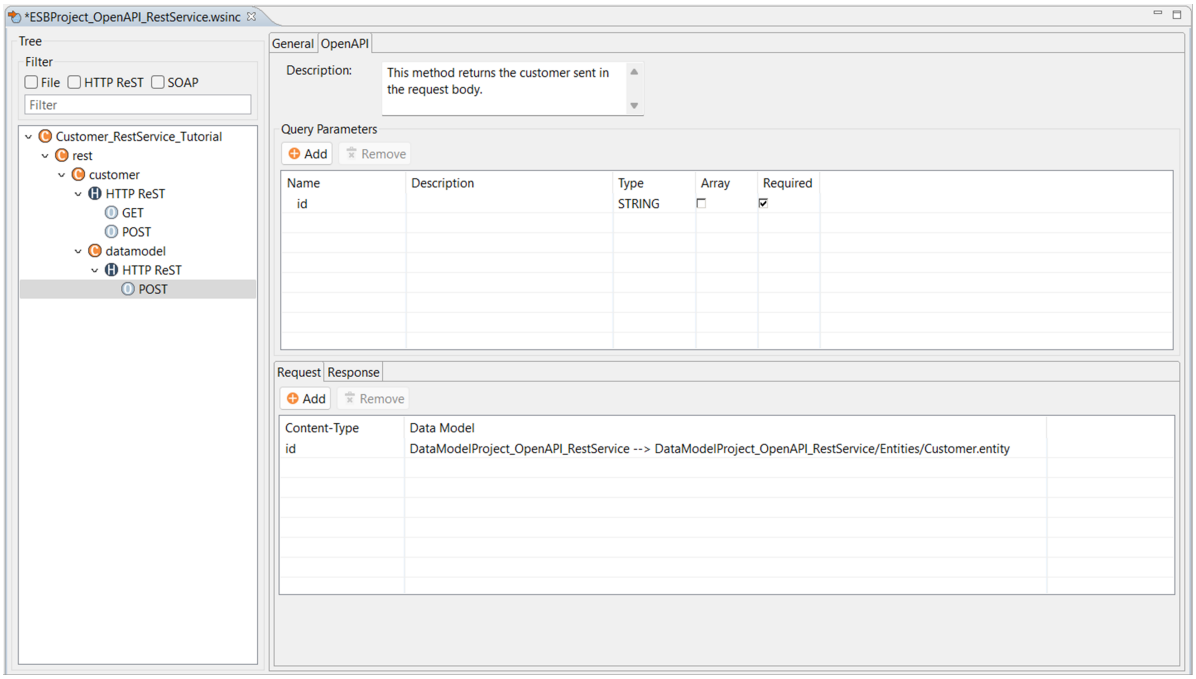
You can configure your ReST service to acquire tabular values from a data model and return tabular values. To do so, you must first create a Data Model Project containing at least one entity in the X4 Designer. Each entity must contain at least one field. For more information, see Working with Data Model Projects.

> ⓘ The data acquisition from a Data Model Project is possible for the `POST`, `PUT`, and `DELETE` met hods.

5.15.3.2 Request with a Data Model

Perform the following steps in the **OpenAPI** tab in order to specify that the input data for a Technical Process is to be provided in a tabular form:

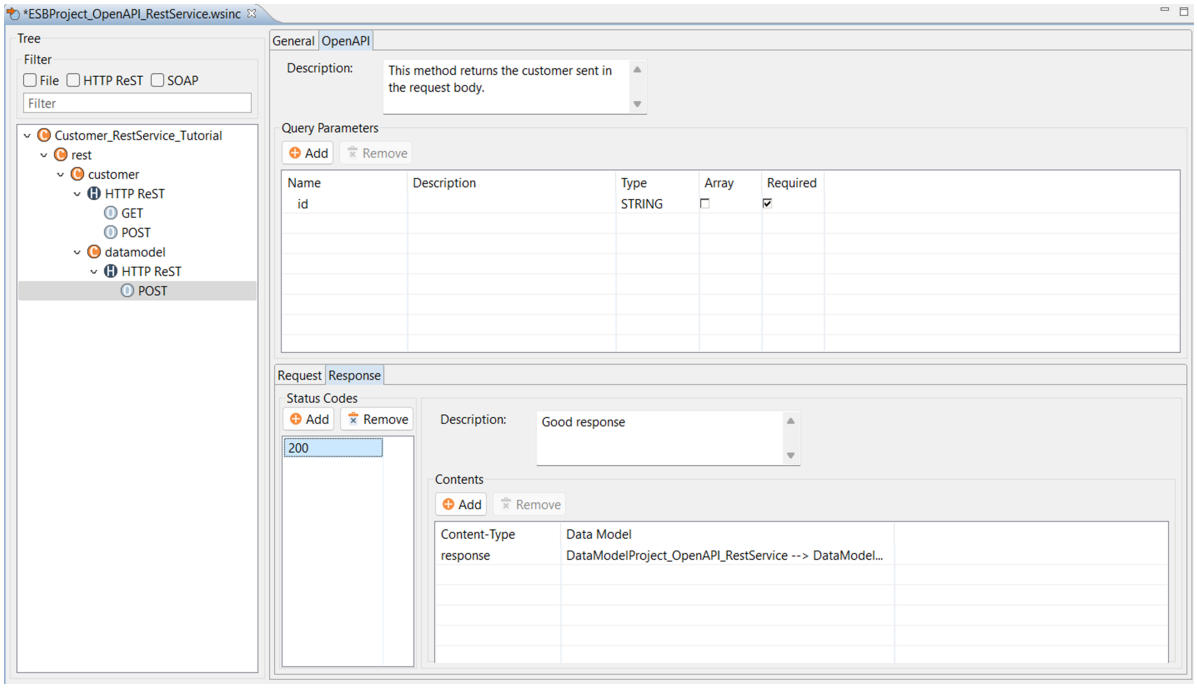
1. Click on the **Add** button in the **Request** tab and enter a name into the **Content-Type** field. Then, click in the **Data Model** field, and choose the button with the three dots.
2. Select the desired data model from the **Data Model** drop-down in the **Select Entities** dialog and check the checkbox for the table whose values you would like to use. The table corresponds to the entity you have created in the Data Model Project.



5.15.3.3 Response with a Data Model

If you also want the output data of a process to be provided in a tabular form, perform the following steps in the **OpenAPI** tab:

1. Choose the **Add** button in the **Status Codes** section and enter the relevant HTTP status code.
2. Enter a description in the **Description** field.
3. Choose **Add** in the **Contents** section in order to add content to the response.
4. Choose the button with the three dots in the **Data Model** field.
5. Select the desired Data Model Project in the **Select Entities** dialog, check the checkbox for the desired table, and choose **OK**.



5.15.3.4 Setting the Authentication

To set a method as private, select the desired method in the *Webservice Configuration Editor*, and proceed as follows:

1. Uncheck the **Inherit from parent node** checkbox for the **Realm** field.
2. Enter the desired value into the **Realm** field, e.g. `X4Realm`.

5.15.3.5 Testing the ReST Service

To open the configured API in the browser, proceed as follows:

1. Call the http://localhost:8080/X4/httpstarter/ReST/[ESB_CONSTANT_VALUE]/openapi.html URL. Replace **[ESB_CONSTANT_VALUE]** with the name of your project. Alternatively, you can right-click on the method operation and choose **Copy OpenAPI URL to Clipboard** from the context menu. Then, insert the copied URL in the browser.
2. Expand the entry for a method under **Service**, and choose **Try it out**.

> ⚠ If your ReST service contains parameters, you must enter those parameters under **Parameter** before clicking on **Execute**.

> ⚠ If your ReST service contains a Data Model Project, the data is contained in the request body. You must enter the required value before clicking on **Execute**.

3. Choose **Execute**.

   The result of the operation is displayed under **Responses**.

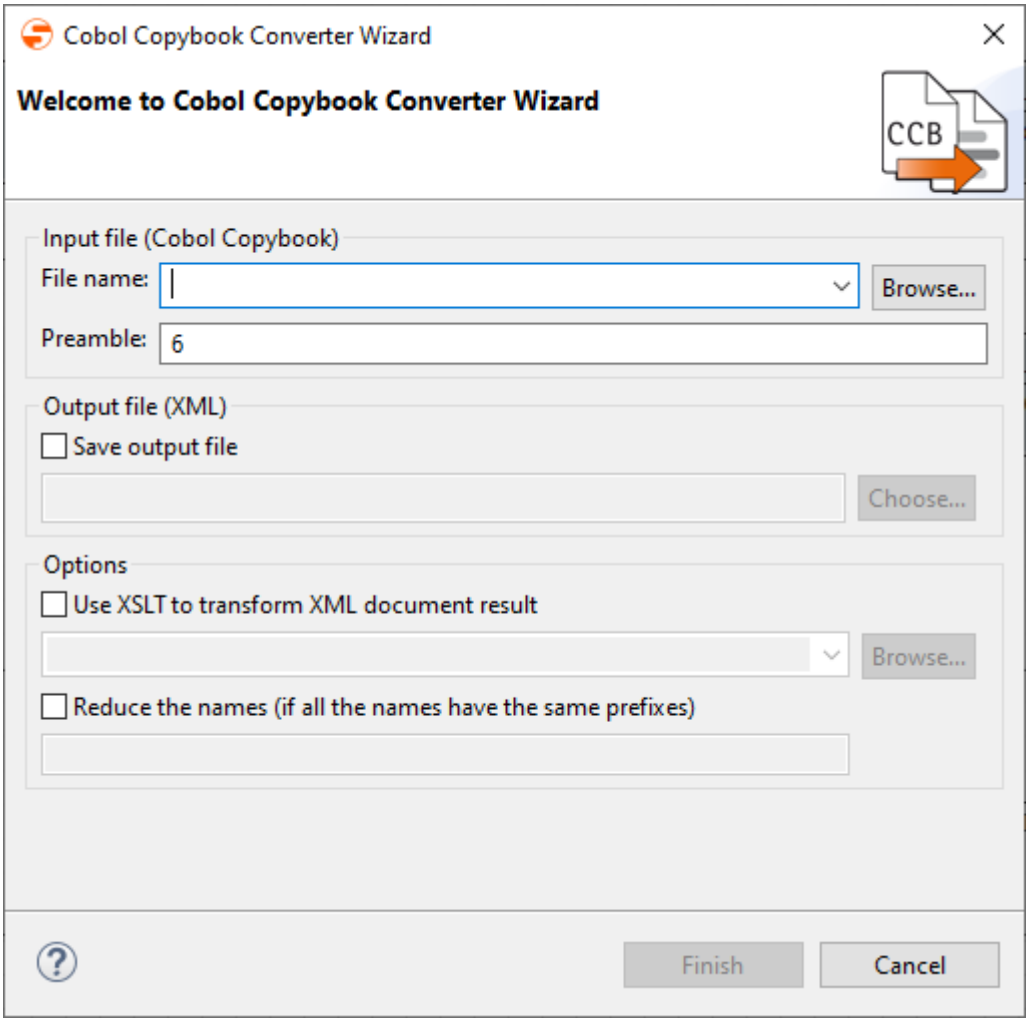
5.15.3.6 Testing a ReST Service with Authentication

If an authentication has been set for a ReST service, the method is marked with a lock icon on the right side. When you open the method, choose the **Try it out** button, enter a parameter (if required), and choose **Execute.** a dialog opens where you must enter your user name and password. Once you have entered your credentials, the method is executed and the response is displayed.

5.16 COBOL Copybook Converter Wizard

With the *COBOL Copybook Converter Wizard* you can convert a text file in *COBOL Copybook* format into an XML document and apply an XSLT transformation of the XML data in the same step. If all names have the same prefixes, you can also reduce all names with the same prefix in the result XML document.

ⓘ To call the *COBOL Copybook Converter Wizard* , click **Project > Cobol Copybook Converter...**.



5.16.1 Requirements

For the wizard to run correctly, the folder `<X4>\X4DB\999\CCB2XML` on the *X4 server* must be complete.

5.16.2 Functions of the COBOL Copybook Wizard

File name	Path and file name of the *COBOL Copybook* file in the repository
Preamble	Preable of the *COBOL Copybook* file (default: *6*)
Display result of conversion	Show preview of the created XML document in the internal browser
Save output file	Save result in a file whose repository path can be selected via **Browse** .
Use XSLT to transform ...	Apply an XSL tranformation to the generated XML document before saving.
Reduce the names ...	Reduce all names with the same prefix

5.17 Extracting the SessionID of a Web App

When a user opens a Web App, the user data can be read out in a Technical Process using the **User Identity Adapter**.

The read user data contains the `SessionID` that the user uses for this session. A `SessionID` is a unique session identifier. The `SessionID` can be used to uniquely associate the user and the session. This allows to place data in the session or access data from the session. The SessionID of a Web App can be read from the so-called **JSON Web Token**.

The `SessionID` are included in the output XML of the **User Identity Adapter**.

For more information, see section User Identity Adapter.

5.18 Read out groups of a user

When a user opens a Web App, the user data can be read out in a Technical Process using the **User Identity Adapter**.

The read out user data contains, among other things, the groups that are assigned to the user. The groups are contained in the output XML of the **User Identity Adapter**.

For more information, see section User Identity Adapter.

5.19 Read out claims of a user

When a user opens a Web App, the user data can be read out in a Technical Process using the **User Identity Adapter**.

The read out user data contains, among other things, the claims of the user. The claims are contained in the output XML of the **User Identity Adapter**.

For more information, see section User Identity Adapter.

6 Debugging and Running Processes

Learn how to validate, debug and execute Technical Processes and Business Processes on the X4 Server.

6.1 Debug Technical Processes

During process development, you can check your Technical Processes and under certain conditions also Business Processes for syntactic errors of the single process modules directly in the Process Editor.

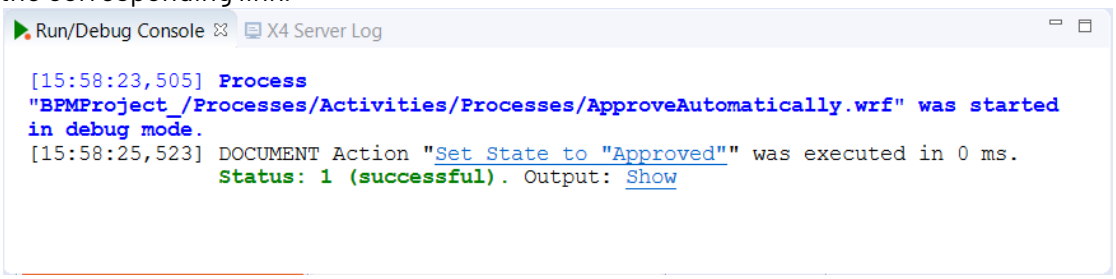
> Requirements
>
> - If required, repository files are linked to process modules.
> - All mandatory parameters and properties are fully parameterized for all process modules.
> - For business processes: The business process was not created in free-flow mode.

1. If desired, set debug breakpoints to automatically stop debugging at critical process steps:
   - Click the desired process module in the process diagram.
   - Click **Toggle Breakpoint** in the toolbar.
2. Click **Start Debugging (F4)** to start the debug mode. To start a parallel debug session, click **Start Debugging** again.

   > You can start a parallel debug session for all Technical Processes.
   > In the **Debug Sessions** view, you can see all active debug sessions. In the **Debug Sessions** view, double-click the desired Technical Process to switch to the debug session.
   > Active debug sessions are marked in gray. New debug sessions connected to the server are marked in orange.

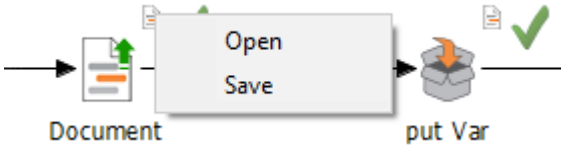
3. Execute the next step.
   a. Click **Step Into (F5)** to execute and display the next single step. The debugger jumps into subprocesses. Each single step in the subprocess must be executed.
   b. Click **Step Over (F6)** to execute and display the next single step. The debugger skips subprocesses, but stops when a breakpoint is reached. The result of the subprocess is displayed as output.
   c. Click **Step Return (F7)** to execute and display all steps in a subprocess to the end or breakpoint. The debugger then jumps to the main process.
   d. Click **Resume (F8)** to execute and display all steps to the end or breakpoint.
   e. Click **Resume without debugging (F10)** to stop debugging and execute all steps. Breakpoints are skipped.
   f. Right-click an element in a process. In the context menu, click **Run to element** to execute and display all steps up to the element or breakpoint.
   For each executed process step, an entry is displayed in the **Run/Debug Console** view. Error messages are highlighted in red, successful status messages are green. In addition, a temporary output file is created for each step, which you can view by clicking

the corresponding link.



Additionally, the respective output file is attached to the corresponding process component in the process model. The output file can also be saved via the context menu.

> If the result contains content larger than 1 MB, a message is displayed and the result can be downloaded via the dialog.



> The Output view and Quick Watch view provide additional capability when debugging:
>
> - **Output**: The results of each step can be viewed during debugging; large files can be downloaded via a download button.
> - **Quick**: placeholders, dynamic parameters and context variables can be viewed during debugging; large files can be downloaded via a download button.
>
> Both views can be accessed via the **Debug > Output** or **Quick Watch** menu or via the respective icons (Output) and (Quick Watch) in the toolbar.

4. With **Terminate (STRG+F2)** you can end the debug mode manually, if necessary.
   If a technical error occurs, the debug mode is automatically terminated.

> For further information, please visit the following sections:
>
> - Execute debug session without debugging
> - Terminate debug session
> - Attach debugger with server

6.1.1 Execute debug session without debugging

You can execute and terminate one or more debug sessions without debugging.

1. Activate checkbox to select the process.
2. Click **Resume without debugging** in the **Debug sessions** view to stop debugging and execute all steps. Breakpoints are skipped.

For further information, refer to section

- Terminate debug session
- Attach debugger with server

6.1.2 Terminate debug session

You can terminate one or more debug sessions without debugging.

1. Activate checkbox to select the process.
2. Click **Terminate** in the **Debug sessions** view to terminate debugging.

For further information, refer to section

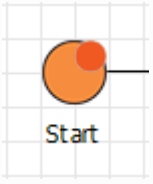
- Execute debug session without debugging
- Attach debugger with server

6.1.3 Attach debugger with server

The server can only be attached to one X4 Designer at a time.
As soon as the server sends data to the debugger and a breakpoint is reached, the X4 Designer starts blinking.

To debug processes with requests or data received from the server, the debugger must be attached to the server. Requests and data from the server are used for debugging, a buffer file is not needed.

To see the data sent by the server, set a breakpoint at the start module.



1. Click **Attach Debug** to attach the debugger to the server.
2. Click **Detach Debug** to detach the debugger from the server.

For further information, refer to section

- Execute debug session without debugging
- Terminate debug session

6.2 Executing Processes

You can start Technical Process and Business Processes directly from within the X4 Designer. The processes are executed on the X4 Server, and no debug information will be displayed in the server log.

To execute a Technical Process or Business Process, you have the following options:

Start opened process	Click **Run Without Debugging** in the toolbar to start the process.
In the Repository Navigator	Right-click on the process, and select **Run Without Debugging** from the context menu.
As subprocess	Subprocesses are started automatically by the parent Technical Process.
Scheduled execution	Start processes at scheduled times with the Scheduler.

6.3 Status Values for Processes and Process Steps

During debugging and when executing a process, the X4 server provides status values for each process step and for executed processes. Any whole numbers (positive, negative, 0) are displayed as status values.

Note: Check the status values of the adapter you are using in the process to ensure that the adapter's status values do not collide with other status values in the process.

Recommendation: Use numbers greater than 1000 for user-defined status values to avoid possible overlaps with other status values.

Number	Label	Meaning
`-999`		Error status of the engine Occurrence: • This error occurs with the engine problem `No branch found to continue` • The process cannot be started because it is broken

Number	Label	Meaning
-998		The execution of the process was not carried out due to licence reasons. Possible reasons: • no licence • damaged licence • expired licence • licence does not match the server version
-1	error	Error during execution of the process component (details are displayed in the server log). The PID (process instance ID) displayed: The process is "Out of Service"
0	successful without result	The process component has been executed and there is no result document or the result is zero (empty).
1	successful	The process component was executed without syntactical errors.
997		The process was abruptly stopped (terminated) as planned (comparable to an uncaught exception in Java). Causes the immediate termination of the entire process.
998		The process was (successfully) stopped and saved in a save point. Causes the immediate termination of the entire process.
999		Status ID of process component not yet executed.
(any other negative integer)		Individual error status set by *ConditionState* mapping; ideally 4-digit; see Set status dynamically with ConditionState
(any other positive integer)		Individual result status set by *ConditionState* mapping or by an adapter with variable status values; ideally 4-digit; see Set status dynamically with ConditionState

7 Scheduler

The scheduler interface provides an overview of all processes for which a scheduled execution has been defined. In addition, the interface also offers the possibility to configure new time-controlled executions as well as to edit or delete already existing configurations.

When defining schedules, a differentiation is made between the following execution options:

- **Startup**: Execution of a process at server startup
- **CRON**: Execution of a process according to certain cron patterns

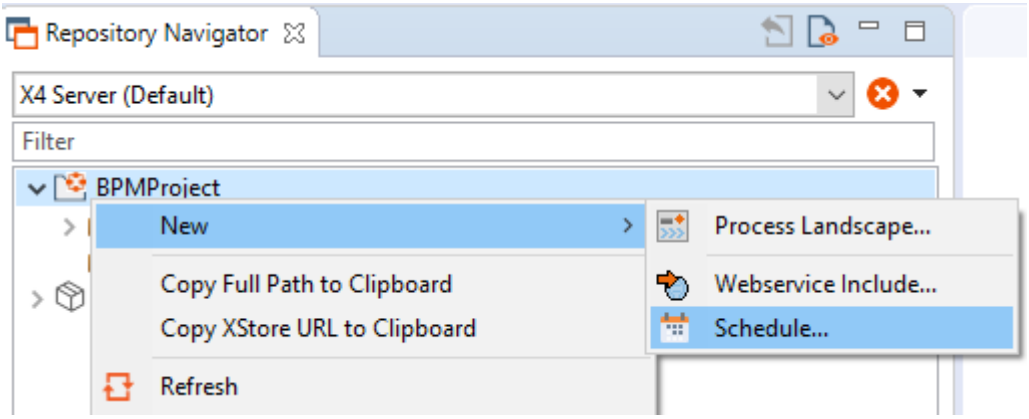
The main functions of the *scheduler* are:

- Overview of existing schedules
- Create/edit any number of execution points and intervals per process

7.1 Create scheduler

- The scheduler can only be created in **BPM** and **ESB** projects.
- Only one scheduler can be created in a project.

1. In the Repository Navigator view, right-click a project.
2. In the context menu select **New> Schedule**



.The scheduler is created and opened.

For further information please visit the section

- Execute processes scheduled
  - Server start
  - Cron

7.2 Execute processes scheduled

The following execution options can be differentiated for the time-controlled execution of X4 processes:

- Execution of a process at server startup
- Execution of a process according to certain Cron patterns

7.2.1 Create new scheduled execution

To create a scheduled execution, proceed as follows:

1. In the scheduler `.schedule` click the desired execution option.
   a. Cron to execute processes according to specific Cron patterns.
   b. Startup to execute processes at server startup.
2. Define the desired configuration.

   For the configuration options of the different types of execution, see:
   - Server start
   - Cron
3. Click **Save** to save the execution.

The ReST interface documentation can be retrieved under http://localhost:8080/X4/X4Api/ in OpenAPI 3.0 format.
To use the ReST interface the Swagger Editor can be used for instance.

7.2.2 Server start

New configurations for executions at server startup can be created using the **Startup** button. A number of options and parameters are available. These are explained in more detail below.

Options

Name	Freely selectable name for execution
Process	Path to the process in the `X4DB` *Example*: `Processes/Activities/Processes/Process.wrf` Click **Browse...** to select a process. ⚠ • The process must be in the same project as the scheduler. • The process must not be a sub-process of another process.

Active	Activates the scheduler
Start date	Sets the start date of the scheduler in `jjjj-mm-tt hh:mm:ss` format *Example:* `2021-06-01 13:00:00`
End date	Sets the end date of the scheduler in `jjjj-mm-tt hh:mm:ss` format *Example:* `2022-08-01 23:00:00` The end date must be in the future.

Parameters

In the **Parameters** section you can define any number of parameters for the run configuration.

	Add row
	Remove row
Key	Name of the parameter (any string, ideally `NCName`).
Value	Optional parameter value

7.2.3 Cron

New configurations for executions at server startup can be created via the **Cron** button. A number of options and parameters are available for this. These are explained in more detail below.

Options

Name	Freely selectable name for execution
Process	Path to the process in the `X4DB` *Example*: `Processes/Activities/Processes/Process.wrf` Click **Browse...** or **drag and drop** a process into the field to select a process. ⚠ • The process must be in the same project as the scheduler. • The process must not be a sub-process of another process.

Active	Activates the scheduler
Generate Cron expression	Generates a Cron expression **Possible values:** • `Minutes` • `Hourly` • `Daily` • `Weekly` • `Monthly` • `Yearly` In the Source tab you can edit the Cron expression manually. Generating a cron expression is not possible if the cron expression was edited manually.

Parameters

In the **Parameters** section you can define any number of parameters for the run configuration.

	Add row
	Remove row
Key	Name of the parameter (any string, ideally `NCName`).
Value	Optional parameter value

8 Process Monitoring

In a file for the monitoring of profiling data (`.profiling`), you can define whether processes within your project should be logged at runtime. In this file, you can define exactly which processes are included or excluded from logging.

The `.profiling` file only needs to be created if you want to exclude specific processes from profiling.

To create a file for the monitoring of profiling data

1. In the **Projects** view, right-click a project.
2. Select **New** > **Monitoring** in the context menu.
   The monitoring configuration file is created and opened.

⚠ Only one `.profiling` file can be created in a project.

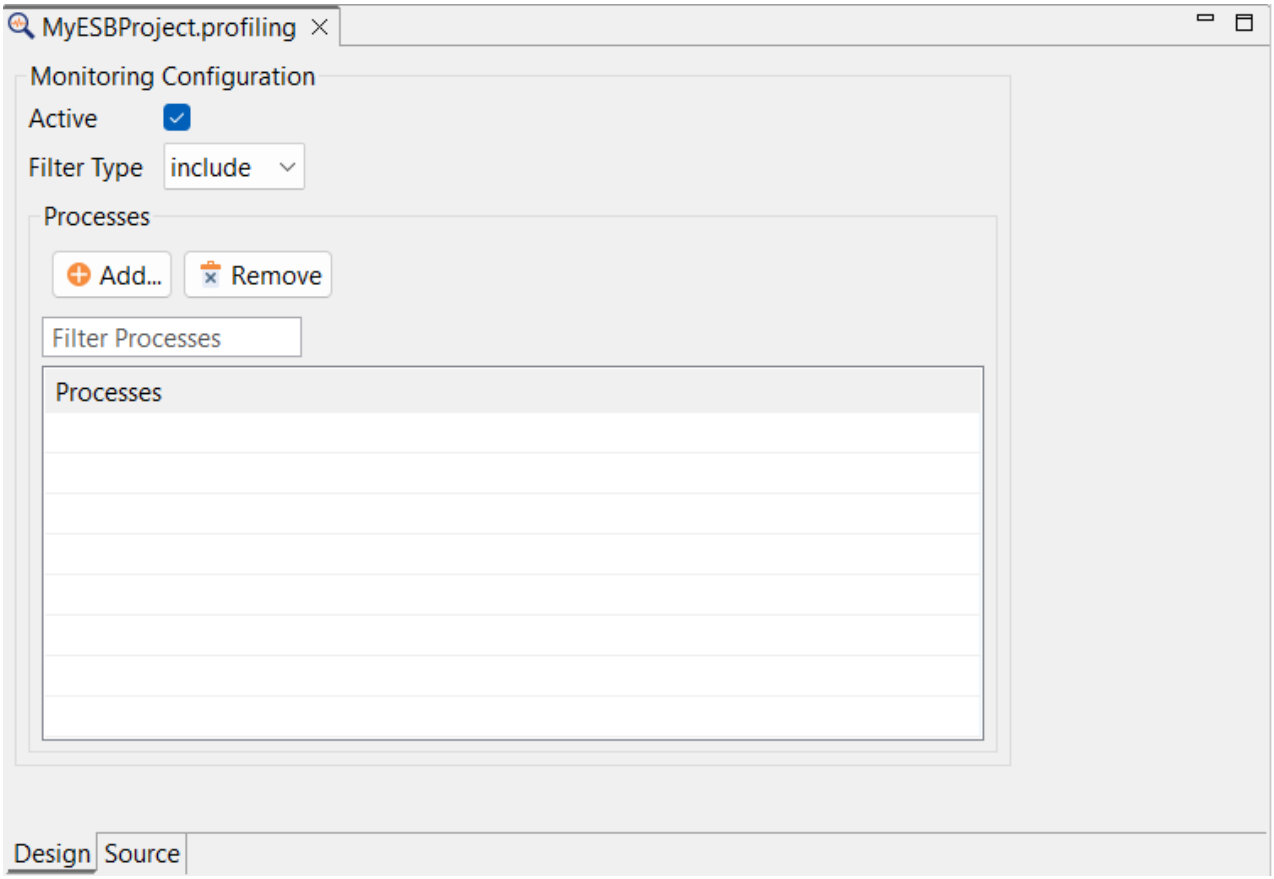
- You can configure the monitoring of profiling data (`.profiling` file) in the **Design** view and in the **Source** view. Both options are described on the following pages. To access the desired view, click **Design** or **Source** at the bottom of a `.profiling` file.

✓ **Additional information:**

- Configuring the Monitoring of Profiling Data in the Design View
- Configuring the Monitoring of Profiling Data in the Source View

8.1 Configuring the Monitoring of Profiling Data in the Design View

When you configure the monitoring in the **Design** view, you will see all available settings in the **Monitoring Configuration** window.



You can make the following settings in the **Monitoring Configuration** window:

Active checkbox	Use this checkbox to specify whether the logging of processes is enabled or disabled. By default, the checkbox is selected.
Filter Type drop-down	Use this drop-down list to specify whether the stored processes should be included or excluded from logging. Available options: **include**: The stored processes are explicitly covered by the logging. **exclude**: he stored processes are explicitly excluded from the logging.
Processes pane	In this pane, you can **Add** and **Remove** processes. You can also specify filter criteria for added processes. For processes you add here, you can specify whether these processes should be included or excluded from logging.

8.2 Configuring the Monitoring of Profiling Data in the Source View

The root element of the profiling file i s `<Profiling>`. The element can contain individual processes of the project that should be explicitly covered or excluded from logging.

The `<Profiling>` element can have the following attribute:

Attribute	Description
`active`	Defines whether logging of processes is enabled or disabled. **Possible values:** • *`true`*: The logging of the processes is enabled (default). • *`false`*: The logging of the processes is disabled.

The `<Profiling>` element contains the `<Processes>` child element, which may contain the following attribute:

Attribute	Description
`filterType`	Defines whether the stored processes should be included or excluded from logging. **Possible values:** • *`include`*: The stored processes are explicitly covered by the logging. • *`exclude`*: The stored processes are explicitly excluded from logging.

The `<Processes>` element may contain `<Process>` child elements if one or more processes are to be explicitly included or excluded from logging.

The `Process` element contains the following attribute:

Attribute	Description
`path`	*Required*. Contains the path to the Technical Process to be included or excluded from the logging. **Possible values:** Repository path to a Technical Process (e.g., `Processes/Process.wrf`)

8.2.1 Example of `.profiling` file

In the following example, the Technical Processes `Process_B.wrf` and `Process_C.wrf` are excluded from logging.

```
<?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 Working with Adapter Projects

Adapter projects allow to develop new adapters for the X4 BPMS using the various tools provided within the X4 BPMS, see Developing Adapters with the X4 BPMS.

> To use this X4 BPMS feature an ADK license is required.

10 X4 Adapters Overview

10.1 About X4 Adapters

The X4 BPMS provides an extensive collection of standard adapters and connectors to integrate applications and data into Business Processes. Thus, the X4 BPMS follows a modular approach. Adapters can be integrated as components into Technical Processes and allow to establish data connections, for example as a connector for Oracle databases, to transfer data using various protocols such as HTTP and JMS, or to transform and convert different data and data formats.

This approach allows heterogeneous IT landscapes to be transferred into a service-oriented architecture.

X4 Adapters feature:

- Plug & play capability
- Continuously growing number of adapters, currently more than 200
- Adapters for all standard protocols, formats and systems
- Easy development of own adapters with the X4 Adapter Development Kit (ADK)
- Easy use within the X4 Designer via drag & drop
- Configuring instead of programming

10.2 Big Data

- Amazon S3 DynamoDB Connector
- Apache Cassandra Connector
- Google Cloud Storage BigTable Connector
- Microsoft Azure Data Lake Store Connector

10.2.1 Amazon S3 DynamoDB Connector

`Amazon S3 DynamoDB Connector`: Amazon S3 DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. The adapter stores data in Amazon S3 DynamoDB Storage allowing to use the X4 BPMS for persisting data in cloud environments.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `CreateTable`: Creates a table with table name and primary key **Required parameters:** `tableName`, `partitionKey`, `partitionType`. Für sortierbare Schlüssel zusätzlich `sortKey`, `sortType`. Für AutoScaling zusätzlich `scalingMinCapacity`, `scalingMaxCapacity`, `scalingTargetValue`, `scalingInCooldown`, `scalingOutCooldown`, `scalableType`, `arnRole` • `UpdateTable`: Updates an existing table modifing the Read/Write Capacity Units. **Required parameters:** `tableName`, `partitionKey`, `readCapacityUnits`, `writeCapacityUnits` • `ListTable`: Lists all tables. • `EnableAutoScaling`: Enables AutoScaling for an existing table according to `scalableType` (Read/Write). **Required parameters:** `tableName`. `scalingMinCapacity`, `scalingMaxCapacity`, `scalingTargetValue`, `scalingInCooldown`. `scalingOutCooldown`, `scalableType`, `arnRole` • `DisableAutoScaling`: Disables AutoScaling for an existing table according to `scalableType` (Read/Write). **Required parameters:** `tableName`, `scalableType` • `PutItem`: Receives an XML input with the structure of the item to be put. **Required parameters:** `tableName` • `BatchPutMultipleItems`: Inserts multiple items in a single call. **Required parameters:** `tableName` **Required parameters** An input XML file containing the elements is required for this operation. • `GetItem`: Gets the specific item **Required parameters:** `tableName`, `partitionKey`, `partitionType`, `partitionValue` (must match with the item to get). For sortable keys `sortKey`, `sortType`, `sortKeyValue` are also required. • `BatchGetMultipleItems`: Inserts multiple items in a single call.

	Required parameters: `tableName`, `partitionKey`, `partitionType`, `partitionValue` (must match with the item to get). For sortable keys `sortKey`, `sortType`, `sortKeyValue` are also required. ⓘ An input XML file containing the elements is required for this operation. • `Delete`: Deletes item or table according to the configured parameters. ⓘ If `tableName` and the primary key elements are specified, the `Delete` operation deletes the specific item. If only `tableName` is specified, the `Delete` operation deletes the table. • `BatchDeleteMultipleItems`: Deletes multiple items in a single call. **Required parameters:** `tableName` ⓘ An input XML file containing the elements is required for this operation. • `Query`: Retrieves a subset of items inside a table. Conditions are specified for the various attributes. For more information on the input file, please see the input section. **Erforderliche Parameter:** `tableName`, `partitionKey` ⓘ `partitionKey` is mandatory and the comparator between `partitionKey` and `sortKey` is always "AND". **Possible operators:** =, >, <, >=, <=, !=, BETWEEN, begins_with

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.amazondynamodb.AmazonDynamoDBAdapter*: Main class (default)

`keyID`	Access key ID used to sign programmatic requests to AWS ℹ The credentials are passed by parameters to the adapter. There are different ways to configure them, see How to get credentials keys. **Possible values:** Any string containing the access key Id, e.g. *`AKIAIOSFODNN7EXAMPLE`*
`secretKey`	Secret access key used to sign programmatic requests to AWS ℹ The credentials are passed by parameters to the adapter. There are different ways to configure them, see How to get credentials keys. **Possible values:** Any string containing the secret access key e.g. *`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`*
`region`	Domain r egion **Possible values:** Any string containing the region name, e.g. *`us-east-2`* ℹ Buckets created in a specific region, can not be accessed from a different region, s ee Regions and Endpoints.

`tableName`	Table name
	
	Possible values: Any string containing the table name
	
	The following naming rules apply to table names:
	• All names must be UTF-8 encoded and are case-sensitive
	• Table names and index names must be between 3 and 255 characters long, and may contain only the following characters:
	  • `a-z`
	  • `A-Z`
	  • `0-9`
	  • `_` (underscore)
	  • `-` (dash)
	  • `.` (dot)
	• Attribute names must be between 1 and 255 characters long

`partitionKey`, `partitionType` and `partitionKeyValue`	Items of the table primary key
	
	Possible values:
	
	• `PartitionKey`: Name of the partition key
	• `PartitionType`: Partition type
	  • **S**: the attribute is a string
	  • **N**: the attribute is a number
	  • **B**: the attribute is of the type Binary
	• `PartitionKeyValue`: Item value; Is required for the operation `GetItem`.

`sortKey`, `sortType` and `sortKeyValue`	Items of the table primary key
	
	Possible values:
	
	• `SortKey`: Name of the sortable key
	• `SortType`: Sorting type
	  • **S**: the attribute is a string
	  • **N**: the attribute is a number
	  • **B**: the attribute is of the type Binary
	• `SortKeyValue`: Item value; Is required for the operation `GetItem`.

`readCapacityUnits`, `writeCapacityUnits`	Read/Write Capacity Units **Possible values:** • Integer Capacity Units
`scalableType`	Defines the scalable type for AutoScaling operation. **Possible values:** • *`ReadCapacity`* • *`WriteCapacity`*
`autoScaling`	Defines if the new table is created by using AutoScaling. In case of a check, all scaling parameters must be specified. **Possible values:** • *`true`* • *`false`*
`scalingMinCapacity`, `scalingMaxCapacity`, `scalingTargetValue`, `scalingInCooldown`, `scalingOutCooldown`	Defines scaling parameters for AutoScaling operation. **Possible values:** • For more information, please see AutoScaling.

`arnRole`	ARN of the role authorized to handle AutoScaling operation for the resource of tables. **Possible values:** • For more information, please see Amazon Resource Names (ARN).
`consistentRead`	DynamoDB uses eventually consistent reads unless it is specified otherwise. Read operations such as `GetItem` and `Query` provide a `ConsistentRead` parameter. If this parameter is set to *`true`*, DynamoDB uses strongly consistent reads during the operation.

Status values

`1`	The operation was executed successfully
`0`	Operation `Delete`: • the item does not exist • the table does not exist
`-1`	An error occurred during the operation's execution

Input

The adapter expects a specific XML structure as input. The structure depends on the operation used.

- `PutItem` operation:

  Expected structure for 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>
```

  In case of the operation `PutItem` there are no restrictions on element names. However the attributes `name`, `value` and `type` are required.

- `BatchGetMultipleItems` operation:

Expected structure for operation BatchGetMultipleItems

```
<?xml version="1.0" encoding="UTF-8" ?>
<PrimaryKeyValues>
    <PrimaryKey partitionKey="2" SortKey="Carlos" />
    <PrimaryKey partitionKey="5" SortKey="Luis" />
</PrimaryKeyValues>
```

Primary keys can be specified for different objects. If the object matches a table where only `partitionKey` is defined, only `partitionKey` may be specified, otherwise `partitionKey` and `SortKey`.

- `BatchPutMultipleItems` operation:

Expected structure for 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>
```

Several items can be specified. In the items, attributes of type Boolean (BOOL), String (S), Number (N), Map (MAP), List (LIST) and Binary (B) can be specified (see example). The output of this operation is the same XML file. This can be useful for further operations.

- `BatchDeleteMultipleItems` operation:

Expected structure for 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>
```

This input specifies the primary keys of the elements to be deleted. If the primary key consists only of `partitionKey`, only `partionKey` is needed. If the primary key consists of `partitionKey` and `SortKey`, both keys are needed to make the item to be deleted unique. The output of this operation is also the input.

- `Query` operation:

Expected structure for 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>
```

Parent element `Query` consists of three elements:

- `KeyCondition`: Only to specify `partitionKey` and if present `SortKey`, meeting following conditions:
  - Comparator between `partitionKey` and `SortKey` is always "AND".
  - `partitionKey` is mandatory and always used with "=".
  - If `SortKey` is used, following operators can be used: =, >, <, >=, <=, !=, BETWEEN, begins_with.
  - `partitionKey` and `SortKey` must not match with reserved words for AWS DynamoDB.
- `OtherFilters`: Specifies conditions for the rest of attributes that are not part of the primary key. The attributes must not match with reserved words for AWS DynamoDB .
- `QueryParameters`: Specifies the mapping between used parameters and their corresponding values. The type of every parameter must be specified.

Output

The adapter returns different kind of documents depending on the executed operation:

- `GetItem` operation:

Sample output for 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:

Sample output for 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>
```

For each element, an `Item` element with fields for the primary key and the associated values is created. Within the `Item` element is a list of the associated attributes.

- `ListTables` operation:

Sample output for operation ListTables

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
    <sampleItems />
</Result>
```

Parent element `Result` is provided that contains one element for each table declared in the account.

- `Query` operation:

Sample output for 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>
```

Parent element `Result` is delivered. Attribute `hasMorePages` specifies if the result has more pages to display or not. This is useful for large responses. For each element that matches the query conditions, an item element is created that contains all attributes.

10.2.2 Apache Cassandra Connector

`Apache Cassadra Connector`: Connects with a Apache Cassandra database.

Properties

`Operation`	Defines the Operation executed by the adapter. **Possible values:** • `Execute`: Executes the statement specified in the input document. • `Terminate`: Closes and removes the connection pooling with all open connections

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.apache.cassandra.ApacheCassandraConnectorAdapter*: Hauptklasse (default)
`host`	Host within the cluster `Cluster cluster1 = Cluster.builder().addContactPoints("172.30.210.11","172.30.210.12").build();` Multiple IPs or host names can be added that correspond to different servers running as an Apache Cassandra executable within the module.
`port`	Native protocol port
`login`	Login if the cluster requires credentials. **Possible values:** Any string containing the login, e.g. *`myLogin`*.
`password`	Password if the cluster requires credentials. **Possible values:** Any string containing the password, e.g. *`myPassword`*.

Status values

`1`	The operation was executed successfully.
`-1`	An error occured during the operation's execution.

Input

The adapter expects a specific XML document as input:

Sample 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>
```

The input xml document must comply with the following rules:

Element / Attribute	Description
`resulttag`	*Required* attribute of the element `Preparedstatement`.
`timeZone`	*Optional* attribute of the element `Preparedstatement`. Defines the time zone corresponding to `java.util.TimeZone` (e.g. *GMT+2*). By default, the UTC time zone is used for dates.
`Cql`	Element within the element `Preparedstatement`. Contains the CQL statement. Can be used any number of times within `Preparedstatement` verwendet werden. ⓘ If characters that are preset in XML are used within the CQL statement, then the SQL statement must be escaped with a CDATA block.
`Parameters`	Element within the element `Preparedstatement`. May only be used once within `Preparedstatement`. Contains any number of elements `parameter` with parameter values.

Element / Attribute	Description

`parameter`	Element within the element `Parameters`. Can be used any number of times within `Parameters`.
	
	Attributes:
	
	• `index`: Defines the position where the parameter value replaces the `?` within the CQL statement.
	• `type`: Defines the parameter type. The following values are possible with Apache Cassandra:
	  • `TEXT`: UTF-8 encoded text
	  • `VARCHAR`: UTF-8 encoded text
	  • `ASCII`: US-ASCII encoded text
	  • `BLOB`: Hexadecimal encoded binary data without validation
	  • `BOOLEAN`: Boolean values
	  • `BIGINT`: 64 bit integer
	  • `COUNTER`: 64 bit counter
	  • `INT`: 32 bit integer
	  • `SMALLINT`: 2 byte integer
	  • `TINYINT`: 1 byte integer
	  • `DATE`: Datum in the format `yyyy-mm-dd`
	  ⓘ Apacha Cassandra encodes dates as a 32 bit integer representing the number of days since January 1, 1970.
	  • `DECIMAL`: Decimal
	  • `DOUBLE`: 64 bit IEEE floating point value
	  • `FLOAT`: 32 bit IEEE floating point value
	  • `INET`: IP address in IPv4 or IPv6 format.
	  • `TIME`: Time
	  ⓘ Apache Cassandra encodes times as a 64 bit integer representing the number of nanoseconds since midnight.
	  • `TIMESTAMP`: Date and time The format must be defined using the `format` attribute, for example:
	  `<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 in the standard UUID format
	  • `VARINT`: Integer
	  • `MAP`: Array of literals in the JSON format
	  • `LIST`: Collection of one or more ordered elements, e.g.

Element / Attribute	Description

```
<parameter index="n" type="LIST">["tcodd@relational.com","ted.cod
d@relational.com"]</parameter>
```

ⓘ The use of double quotes is required.

- `SET`: Collection of one or more elements, e.g.

```
<parameter index="n" type="SET">["Element 1","Element 1","Element
3"]</parameter>
```

ⓘ The use of double quotes is required.

- `TUPLE`: Group of comma-separated fields. Can contain different simple types:
  - `keySpace`: Name of the key space of the table
  - `table`: Name of the table
  - `column`: Name of the column

ⓘ If the tuple contains date, time or timestamp fields, the following attributes are required:

- `formatWithoutTime`: Required if the tuple contains date fields.
- `format`: Required if the tuple contains timestamp fields.
- `formatTime`: Required if the tuple contains time fields.

Example 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, first 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>
```

In the example above, the tuple field is defined as follows during table creation:
`tupleColl tuple<int, text, varchar, float, time, timestamp, date, inet, decimal, uuid, boolean, ascii, blob>`

Output

The output is an XML document containing the errors and the result of the execution. Depending on the executed operation, the adapter outputs data in different XML structures:

- Batch operations (`INSERT`, `UPDATE`, `DROP`, `CREATE`, ...)

  Example Batch Operations

```
<Result>
    <resulttag_name> Operation executed successfully</resulttag_name>
</Result>
```

- SELECT

  Example 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>
```

  If the field has the data type `MAP`, its value is represented by an object node. If the field has the data type `LIST`, `SET` or `TUPLE`, its value is represented by an array node. If the field is a simple type, its value is represented within the value node.

Example

- CREATE

Example 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

Example DROP

```
<RootElement>
    <Preparedstatement resulttag="drop_Users">
        <Cql>
            <![CDATA[DROP TABLE IF EXISTS videodb.users;]]>
        </Cql>
    </Preparedstatement>
</RootElement>
```

- INSERT

Example 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

Example 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

Example 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

Example 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 Connector`: Enables to connect to Google Cloud BigTable, a highly available NoSQL data store that offloads the database administration.

Properties

`Operation`	Defines the operation executed by the Adapter. **Possible Values:** • `CreateTable`: Defines the operation executed by the Google Cloud Bigtable Adapter. **Required parameters:** `tableName`, `columnFamilyName` • `ListTable`: List all tables for project and instance **Required parameters:** `tableName` , `columnFamilyName` • `PutRow`: Insert rows using values from the XML input document, Using `columnName` and `value` for the `rowKey` attribute **Required parameters:** `tableName` , `columnFamilyName` • `GetRow`: Search a table by `rowKey` passing by parameters and return it in an output XML. **Required parameters:** `tableName` , `columnFamilyName` • `Scan`: Scan a table and return an output XML, according with the parameters. Scan-operation will filter the search. **Required parameters:** `tableName`, `columnFamilyName`, `columnName`, `startRow`, `stopRow`, `scanCached`, `scanResultSize`, `scanTimestamp`, `scanMaxVersions`, `scanCacheBlocks` • `EnableTable`: Enable the table (Table must be in Disable state). **Required parameters:** `tableName` , `columnFamilyName` • `DisableTable`: Disable the Table (Table must be in Enable state). **Required parameters:** `tableName` , `columnFamilyName` • `Delete`: Delete table. If both parameters `rowKey` and `columnName` are set, this operation will only delete a specific column for the chosen row using. **Required parameters:** `tableName`, `columnFamilyName`, `columnName`, `rowKey`

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.google.cloud.bigtable.storage.GoogleCloudBigtableAdapter*: Main class (default)

`serviceAccountFile`	Path to the JSON file that contains the service account key to authenticate to the Google Cloud Storage.
	
	Possible values:
	
	• System path, for example:
	`C:\X4\V_5.9\Server\X4DB\1\ExampleProcess\GoogleCloud\Config\serviceAccountFile.json`
	• X4db url, for example:
	`x4db://1/ExampleProcess/GoogleCloud/Config/serviceAccountFile.json`
	• Xstore url, for example:
	`xstore://ExampleProcess/GoogleCloud/Config/serviceAccountFile.json`

`projectID`	ID of the project. **Possible values:** • ID of Google Cloud project. Can be checked in the Google Cloud console.
`instanceID`	ID of the instance **Possible values :** • ID of Google Cloud instance. Can be checked in the Google Cloud console. Please follow this link for more information on how to create an instance in Google BigTable.
`tableName`	Name of the table. Required parameter for all operations.
`columnFamilyName`	Family name of the columns. Group columns in a unique Column family name.
`rowKey`	Row key of the row. Row identifier to put or search rows.
`columnName`	Name of the column. Used to delete a specific column in a row with `Delete` operation.
`startRow`	Specify a single row to start scanning. **Possible values:** `rowKey`
`stopRow`	Specify a single row to stop scanning, this row will not be shown in the scan results. **Possible values:** `rowKey`
`scanCached`	Modify scanner caching for just this scan, If caching is not set, the caching value of the hosting table is used. **Possible values:** Integer
`scanResultSize`	Specify a maximum result size **Possible values:** Integer ⓘ When both parameters, `scanCached` and `scanResultSize`, are set, single server requests are limited by either number of rows or maximum result size, whichever limit comes first.
`scanTimestamp`	Retrieve columns with a specific timestamp only. **Possible values:** Integer

`scanMaxVersions`	Limit the number of versions of each column to be returned **Possible values:** Integer
`scanCacheBlocks`	Explicitly disable server-side block caching for this scan **Possible values:** `true` / `false`

Status values

`1`	The operation was executed successfully.
`0`	The operation is not complete, but no error occurred.
`-1`	An error occurred during the operation's execution.

Input

- For `PutRow` operation, the adapter expects XML with this structure:

```
<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>
```

> Element names are not restricted, but attributes `columnName`, `value` and `rowKey` are required.

Output

Depending on the executed operation, the adapter returns different kinds of documents.

- `GetRow` operation:

```
<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:

```
<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:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Result>
    <ListTables>
    <Table name="TestTable" />
    </ListTables>
</Result>
```

10.2.4 Microsoft Azure Data Lake Store Connector

`Microsoft Azure Data Lake Store Connector`: Enables communication with first generation Microsoft Azure Date Lakes (based on Apache Hadoop).

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `AppendFile`: Append a file to an already existing file. • `ConcatenateFiles`: Concatenate a specific list of files into a destination file. The source files will be deleted, if the concatenation succeeds. The target file will be created, if it does not exist. Otherwise, the content will be concatenated. • `CreateFolder`: Create a folder. • `DeleteFile`: Delete a file. • `DeleteFolder`: Recursively delete a folder and its complete content. • `EnumerateDirectory`: List all files and folders from a specified folder path. • `GetMetadata`: Get metadata of a file or folder. • `ReadFile`: Read a file. • `RenameFile`: Rename a file. If the target file exists (a file with a same name), the source file will be only renamed, if the parameter `overwrite` is enabled. • `RenameFolder`: Rename a folder. If the target folder exists, the source folder will be moved into the target folder. • `SetPermission`: Set the permissions of the specified file or directory. • `UploadFile`: Create a new file.

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.microsoft.azure.datalake.MicrosoftAzureDataLakeStoreConnector*: main class (default)
`authEndpoint`	*Required*: OAuth 2.0 token endpoint (v1) **Possible values:** String containing a valid endpoint, e.g. *`https://login.microsoftonline.com/<DirectoryID>/oauth2/token`*

`clientID`	*Required*: Client id of the Azure Active Directory application. **Possible values:** String containing the client ID, e.g. *`10925xxx-1111-4444-bd3d-82165c0d3841`*
`clientSecret`	*Required*: Client secret of the Azure Active Directory application. **Possible values:** String containing the secret value.
`accountName`	*Required*: Fully qualified Data Lake Storage account name (FQDN). **Possible values:** String containing the FQDN, e.g. *`<your_account_name>.azuredatalakestore.net`*
`fileName`	File name ⓘ • This parameter is required for the operations `AppendFile`, `ConcatenateFiles`, `DeleteFile`, `ReadFile`, `RenameFile` and `UploadFile`. • This parameter is optional for the operations `GetMetadata` and `SetPermission` **Possible values:** String containing the file name, e.g. *`inputFile.txt`* or *`root/subfolder/inputFile.txt`* ⓘ The string may contain both the complete file path (*`root/subfolder/inputFile.txt`*) or the `fileName` (*`inputFile.txt`*) and `folderPath` (*`root/subfolder`*).
`folderPath`	Folder path; It can start and end with a slash. **Possible values:** • */*: Root folder • Any other valid path, e.g. *`root/subFolder/.../lastFolder`*
`overwrite`	Flag to overwrite existing files. ⓘ This parameter is only valid for the operations `UploadFile` and `RenameFile`. **Possible values:** • `true`: Overwrite the existing file (defined in `fileName`) with the new content • `false`: For the operation `UploadFile` the adapter returns an error, if the file already exists. For the operation `RenameFile` the adapter returns the status `1` but doesn't rename the source file.

`listFilesToConcatenate`	Name of the files to be concatenated (separate by colon). ⓘ This parameter is required for the operation `ConcatenateFiles`. **Possible values:** Existing files (full path names) separate by colon, e.g. *`files/File1.txt:files/File2.txt:otherPath/File3.txt`*
`renamedObject`	New path name of the file or directory to be renamed. ⓘ This parameter is required for the operations `RenameFile` and `RenameFolder`. **Possible values:** Path name, e.g. *`files/newFileName.txt`*
`octalPermissions`	Permissions to be set in unix octal form. For more information on the Octal Permission Representation, see also https://www.tutorialspoint.com/unix/unix-file-permission.htm. *Example*: *`770`* → *`7`* Own user with permission to read, write and to execute; *`7`* Gro up with permission to read, write and to execute; *`0`* Others with no permissions. ⓘ This parameter is required for the operation `SetPermission`.
`maxEntriesToRetrieve`	Maximum number of entries to be retrieved. ⓘ This parameter is optional and will be ignored, if the parameter `entriesRecursively` is enabled. It is useful for the operation `EnumerateDirectory`. ⚠ Note that the server can limit the number of entries retrieved to a number smaller than the number specified. **Possible values:** Integer value greater than `0`.
`startAfterPathName`	The file or directory name after which to begin the enumeration. ⓘ This parameter is optional and will be ignored, if the parameter `entriesRecursively` is enabled. It is useful for the operation `EnumerateDirectory`. **Possible values:** String containing the path name of the file or folder.

`entriesRecursively`	Enables to extract all the entries recursively. If the parameter is enabled `maxEntriesToRetrieve` and `startAfterPathName` are ignored. ⓘ This parameter is optional and useful for the operation `EnumerateDirectory`. **Possible values:** • `true`: All entries of the given folder path will be retrieved recursively. • `false`: All entries of the given folder path will be retrieved, so only direct children will be retrieved.

Status values

`1`	The operation was executed successfully
`-1`	An error occurred during execution

Input

The adapter expects any input document for the operations `UploadFile` and `AppendFile`. The other operations do not require any input.

Output

Depending on the selected operation, the adapter will output various documents:

`AppendFile`

- If the operation was executed successfully (status `1`), the adapter outputs the input document.
- In case of an error (status `-1`), the adapter outputs an XML document with the error description, e.g.

```
<?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`

- If the operation was executed successfully (status `1`), the adapter outputs the input document in case any input document was assigned.
- In case of an error (status `-1`), the adapter outputs an XML document with the error description, e.g.

```
<?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`

- If the operation was executed successfully (status `1`), the adapter outputs the input document.

- In case of an error (status `-1`), the adapter outputs an XML document with the error description, e.g.

```
<?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`

- If the operation was executed successfully (status `1`), the adapter outputs the input document in case any input document was assigned.
- In case of an error (status `-1`), the adapter outputs an XML document with the error description, e.g.

```
<?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`

- If the operation was executed successfully (status `1`), the adapter outputs the input document in case any input document was assigned.
- In case of an error (status `-1`), the adapter outputs an XML document with the error description, e.g.

```
<?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`

- If the operation was executed successfully (status `1`), the adapter outputs an XML document containing a list with all files and folders found, e.g.:

```
<?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"/>
```

> Each `Entry` node corresponds to a file folder. The following attributes characterize each `Entry`:
>
> - `LastAccessTime`: Time of the last access to the file or folder.
> - `LastModifiedTime`: Time of the last modification.
> - `Group`: ID of the group owning the file or folder.
> - `User`: ID of the user owning the file or folder.
> - `Permission`: Unix-style permissions string for this file or folder.
> - `Name`: Name of the file or folder.
> - `FullName`: Complete path of the file or folder.

- In case of an error (status `-1`), the adapter outputs an XML document with the error description, e.g.:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>The folderPath parameter must be informed</Error>
```

`GetMetadata`

- If the operation was executed successfully (status `1`), the adapter outputs an XML document containing the metadata information of the file or directory.

```
<?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>
```

> The root element `Metadata` contains an `Entry` element with the following attributes:
>
> - `Group`: ID of the group owning the file or folder.
> - `User`: ID of the user owning the file or folder.
> - `Permission`: Unix-style permissions string for this file or folder.
> - `Name`: Name of the file or folder.
> - `FullName`: Complete path of the file or folder.
> - `LastAccessTime`: Time of the last access to the file or folder.
> - `LastModifiedTime`: Time of the last modification or folder.

- In case of an error (status `-1`), the adapter outputs an XML document with the error description, e.g.:

```
<?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`

- If the operation was executed successfully (status `1`), the adapter outputs the file that was read:
- In case of an error (status `-1`), the adapter outputs an XML document with the error description, e.g.:

```
<?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`

- If the operation was executed successfully (status `1`), the adapter returns the input document in case any input document was assigned.
- In case of an error (status `-1`), the adapter outputs an XML document with the error description, e.g.:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>The 'renamedObject' parameter must be set for this operation</Error>
```

`RenameFolder`

- If the operation was executed successfully (status `1`), the adapter returns the input document in case any input document was assigned.
- In case of an error (status `-1`), the adapter outputs an XML document with the error description, e.g.:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>The 'renamedObject' parameter must be set for this operation</Error>
```

`SetPermission`

- If the operation was executed successfully (status `1`), the adapter returns the input document in case any input document was assigned.
- In case of an error (status `-1`), the adapter outputs an XML document with the error description, e.g.:

```
<?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`

- If the operation was executed successfully (status `1`), the adapter outputs the file that was uploaded.
- In case of an error (status `-1`), the adapter outputs an XML document with the error description, e.g.:

```
<?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>
```

If the parameter `folderPath` is empty, the file will be created in the account's root folder. If the path set in parameter `folderPath` doesn't exist, it will be created.

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

`AmazonS3 Simple Storage Connector`: Stores data in Amazon S3 Blob Storage allowing to use the X4 BPMS for persisting data in cloud environments.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `CreateBucket`: Creates a bucket. ℹ For the operation `CreateBucket` different configurations are possible. If the parameter `bucketVersion` is set to *ENABLED*, the bucket versioning is enabled. Otherwise the versioning is enabled. If the parameter `publicAccess` is enabled a permission must be defined via the parameter `publicPermission`. **Required parameters:** `bucket`, `bucketVersion`, `publicAccess`, `PublicPermission` • `ChangeBucketVersion`: Changes the bucket's versioning configuration. ℹ New buckets have the status `OFF` by default. Once versioning is enabled, the status can no longer be reset to the status `OFF`. **Required parameters:** `bucket`, `bucketVersion` • `GetAccessControlList`: Lists the grants and permissions for a bucket or file. ℹ If the parameters `bucket` and `file` are specified, the grants and permissions are listed for the file. If only `bucket` is specified, the grants and permissions are listed for the bucket. **Required parameters:** `bucket`, `file` • `AddGrantsPermission`: Adds grants with permissions to a `bucket` or a `file`. ℹ If the parameter `publicAccess` is enabled, a publicly accessible bucket or file is created with the permissions defined in the `publicPermission` parameter. Otherwise, information is taken from the input. If the parameters `bucket` and `file` are specified, the grants and permissions are added to the file. If only `bucket` is specified, the grants and permissions are added to the bucket. **Required parameters:** `bucket`, `file` • `RevokeGrantsPermission`: Revokes all permissions for a grant. ℹ If the parameters `bucket` and `file` are specified, the grants and permissions are revoked for the file. If only `Bucket` is specified, the grants and permissions are revoked for the bucket. **Required parameters:** `bucket`, `file`

- `RevokePublicPermission`: Revokes the public access permissions.
  If the parameters `bucket` and `file` are specified, the public access permissions to the file are revoked. If only `bucket` is specified, the public access permissions to the bucket are revoked.
  Required parameters: `bucket`, `file`

- `AddBucketPolicy`: Adds a bucket policy to the specified bucket.
  Find additional information such as sample bucket policies, permissions in a policy and conditions in a policy.
  Required parameters: `bucket`

- `GetBucketPolicy`: Invokes the bucket policy.
  Required parameters: `bucket`

- `RemoveBucketPolicy`: Deletes the policy for the specified bucket.
  Required parameters: `bucket`

- `Search`: Searches for information in the AmazonS3 Simple storage account and optionally for object metadata.
  If the parameter `includeObjectsMetadata` is selected, metadata information from each object are added to the default response.
  Required parameters: `file`, `includeObjectsMetadata`

- `Upload`: Uploads files to the Amazon S3 Simple storage account.
  Create folder structures using "/", e.g. `Lib/Document/doc.txt`
  Required parameters: `bucket`, `file`, `overwrite`, `path`

- `Download`: Downloads files from the Amazon S3 Simple storage account.
  Required parameters: `bucket`, `file`

- `Copy`: Copies files to the destination bucket.
  To copy a file, permission from the destination bucket is required.
  Required parameters: `bucket`, `file`, `targetBucket`, `targetFile`

- `Move`: Moves files to the destination bucket.
  To move a file, permission from the destination bucket is required.
  Erforderliche Parameter: `bucket`, `file`, `targetBucket`, `targetFile`

	• `Rename`: Renames files in the destination bucket. To rename a file, permission from the destination bucket is required. • `Delete`: Deletes files or buckets. If bucket and file are specified, the specified file will be deleted. If only the bucket is specified, all files inside that bucket and then the bucket will be deleted. • `DeleteMultiFiles`: Deletes multiple files. **Erforderliche Parameter:** `bucket`, `file`

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.amazpns3.AmazonS3SimpleStorageAdapter*: Main class (default)
`keyId`	Access key ID used to sign programmatic requests to AWS The credentials are passed by parameters to the adapter. There are different ways to configure them, see How to get credentials keys. **Possible values:** Any string containing the access key Id, e.g. *`AKIAIOSFODNN7EXAMPLE`*
`secretKey`	Secret access key used to sign programmatic requests to AWS The credentials are passed by parameters to the adapter. There are different ways to configure them, see How to get credentials keys. **Possible values:** Any string containing the secret access key e.g. *`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`*
`region`	Bucket region **Possible values:** Any string containing the region name, e.g. *`us-east-2`* Buckets created in a specific region, can not be accessed from a different region, see Regions and Endpoints.

`bucket`	Bucket name ℹ The bucket must be unique across all existing bucket names in Amazon S3, see Buckets. **Possible values:** Any string containing the bucket name The following rules apply for naming S3 buckets in all AWS Regions. Bucket names... • must be unique across all existing bucket names in Amazon S3 • must comply with DNS naming conventions • must be at least 3 and no more than 63 characters long • must not contain uppercase characters or underscores • must start with a lowercase letter or number • must be a series of one or more labels Adjacent labels are separated by a single period (`.`). Bucket names can contain lowercase letters, numbers, and hyphens. Each label must start and end with a lowercase letter or a number. • must not be formatted as an IP address, e.g. `192.168.5.4` • When using virtual hosted-style buckets with Secure Sockets Layer (SSL), the SSL wildcard certificate only matches buckets that don't contain periods. To work around this, use HTTP or write your own certificate verification logic. We recommend not to use periods (`"."`) in bucket names when using virtual hosted-style buckets.
`bucketVersion`	Bucket versioning to keeping multiple variants of an object in the same bucket **Possible values:** • `OFF`: Versioning is disabled (default) ℹ If versioning for a bucket is enabled, the versioning can no longer be disabled and set to the status `OFF`. Versioning can only be suspended and set to the status `SUSPENDED`. • `ENABLED`: Bucket versioning is enabled • `SUSPENDED`: Bucket versioning is suspended, if the versioning was enabled before
`publicAccess`	Specifies whether buckets or files are publicly accessible or not during the `AddGrantsPermission` operation.

<table>
<tr><td>publicPermission</td><td>Public Access Configuration.

Possible values:

<table>
<tr><th>Permission</th><th>When granted on a bucket</th><th>When granted on an object</th></tr>
<tr><td>READ</td><td>Allows to list the objects in the bucket</td><td>Allows to read the object data and its metadata</td></tr>
<tr><td>WRITE</td><td>Allows to create, overwrite, and delete any object in the bucket</td><td>Not applicable</td></tr>
<tr><td>READ_ACP</td><td>Allows to read the bucket ACP</td><td>Allows to read the object ACP</td></tr>
<tr><td>WRITE_ACP</td><td>Allows to write the ACP for the applicable bucket</td><td>Allows to write the ACP for the applicable object</td></tr>
<tr><td>FULL_CONTROL</td><td>Allows to access the bucket permissions READ, WRITE, READ_ACP, and WRITE_ACP</td><td>Allows to access the object permissions READ, WRITE, READ_ACP, and WRITE_ACP</td></tr>
</table>
</td></tr>
<tr><td>file</td><td>Name of file to be saved, see Objects

The following character sets are generally allowed in key names:

• Alphanumeric characters [0-9a-zA-Z]
• Special characters: !, -, _, ., *, ', (, and)

Possible values: Any string containing the object key name, e.g.

• 4my-organization
• my.great_photos-2014/jan/myvacation.jpg
• videos/2014/birthday/video1.wmv</td></tr>
<tr><td>path</td><td>Path of the file to be uploaded

Possible values: Any valid path, e.g.

• X4DB Path: x4db://1/.....
• XSTORE Pathxstore://
• Full Path: C:/Documents/....</td></tr>
<tr><td>overwrite</td><td>Information whether the file should be overwritten or not

Possible values:

• True: Overwrites the file
• False: Does not overwrite the file</td></tr>
<tr><td>targetBucket</td><td>Name of the bucket where the file will be copied or moved to

Possible values: See parameter bucket</td></tr>
</table>

`targetFile`	Name the file will take when copied or moved to the new path **Possible values:** See parameter `file`
`includeObjectsMetadata`	Information whether object metadata should be included as part of the response **Possible values:** • `True`: Object metadata are included as part of the response • `False`: Object metadata are not included as part of the response

Status values

`1`	The operation was executed successfully
`0`	The operation's execution has not yet been completed or no result has been output
`-1`	An error occurred during the operation's execution

Input

The adapter expects any input document, e.g. binary document or a base-64 document (XML or text).

- Operation `Upload`: If parameter `path` is specified, no input document is required. In all other cases an input document is needed.
- Operation `Download` and `Search`: No input document is required.
- Operation `AddGrantsPermission`:

Sample input for the 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>
```

To add object permissions, the parameters `bucket` and `file` are required. To add bucket permissions, the parameter `bucket` is required and `file` has to be empty.

- Operation `RevokeGrantsPermission`:

Sample input for the operation RevokeGrantsPermission

```
<?xml version='1.0' encoding='UTF-8'?>
<RootElement>
    <Grant id="Canonical User ID">
    </Grant>
</RootElement>
```

To add object permissions, the parameters `bucket` and `file` are required. To add bucket permissions, the parameter `bucket` is required and `file` has to be empty.

- For the operation `AddBucketPolicy` the input must have Json format.
  Find additional information such as sample bucket policies, permissions in a policy and conditions in a policy.
- Operation `DeleteMultiFiles`:

Sample input for the 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>
```

Output

The adapter returns different kind of documents depending on the executed operation:

- Operation `Upload`:
  - If the adapter was executed successfully (status `1`), the adapter returns an XML document.
  - If an error occurred during the adapter's execution (status `-1`), the adapter does not return anything.
- Operation `Search`:
  - If the adapter was executed successfully (status `1`), the adapter returns an XML document with the used filters.

Sample output for 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>
```

- If the adapter was executed successfully (status `1`) and the parameter `IncludeObjectsMetada` is selected, the metadata for each object are output additionally:

Sample output for operation SEARCH including metadata

```
<?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>
```

    - If an error occurred during the adapter's execution (status `-1`), the adapter does not return anything.
- Operation `Download`:
    - If the adapter was executed successfully (status `1`), the adapter returns the downloaded blob (text, XML, binary doc etc.).
    - If the object or bucket do not exist the status `-1` is returned.
- Operation `GetAccessControlList`:

Sample output for the 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 is a highly available NoSQL data store that offloads the work of database administration.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `CreateDomain`: Creates a domain containing the data set. **Required parameters:** `domain` • `DomainMetadata`: Retrieves metadata related to a domain, see official guide. **Required parameters:** `domain` • `DeleteDomain` : Deletes a domain. **Required parameters:** `domain` • `ListDomain`: Lists all domains. • `Put`: Adds or updates an item and its attributes, or adds attribute-value pairs to an already existing item; The item is automatically indexed as it is received. According to the parameter `replaceExistingAttributes`, the content will replace existing attributes or will create new attributes keeping the current ones as well. **Required parameters:** `domain`, `replaceExistingAttributes` • `BatchPutAttributes`: Adds or updates up to 25 `Put` operations in a single call. Items are automatically indexed as they are received. According to the parameter `replaceExistingAttributes`, the content will replace existing attributes or will create new attributes keeping the current ones as well, see official guide. **Required parameters:** `domain`, `replaceExistingAttributes` • `GetItem`: Retrieves an item and all or a subset of its attributes and values. **Required parameters:** `domain`, `itemName` • `DeleteItem`: Deletes an item, an attribute, or an attribute value. **Required parameters:** `domain`, `itemName` • `BatchDeleteAttributes`: Deletes up to 25 `Delete` operations on a single call, see official guide. **Required parameters:** `domain` • `Select`: Queries the data set in the familiar *“`select target from domain_name where query_expression`”* syntax. Supported value tests are: `=`, `!=`, `like`, `not like`, `between`, `is null`, `is not null`, and `every ()`. *Example:* `select * from mydomain where every(keyword) = 'Book'`. Order results using the `SORT` operator, and count items that meet the condition(s) specified by the predicate(s) in a query using the `Count` operator. • `StoreAmazonS3ObjectMetadata`: Stores the metadata related to an AmazonS3 object in a specific domain set in the parameter `domain`. **Required parameters:** `domain`, `replaceExistingAttributes`

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.amazonsimpledb.AmazonSimpleDBAdapter*: Main class (default)
`keyId`	Access key ID used to sign programmatic requests to AWS ℹ The credentials are passed by parameters to the adapter. There are different ways to configure them, see How to get credentials keys. **Possible values:** Any string containing the access key Id, e.g. *AKIAIOSFODNN7EXAMPLE*
`secretKey`	Secret access key used to sign programmatic requests to AWS ℹ The credentials are passed by parameters to the adapter. There are different ways to configure them, see How to get credentials keys. **Possible values:** Any string containing the secret access key e.g. *wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY*
`region`	Domain region **Possible values:** Any string containing the region name, e.g. *us-east-2* ℹ Buckets created in a specific region, can not be accessed from a different region, see Regions and Endpoints.
`domain`	Domain name identified by a unique key (name) Domains are represented by domain worksheet tabs at the bottom of the spreadsheet and are similar to tables containing similar data. The name must be between 3 and 255 characters and can contain the following characters: `a-z`, `A-Z`, `0-9`, '_', '-', and '.'. ℹ You can execute queries against a domain, but cannot execute queries across different domains.
`itemName`	Item name Items are represented by rows and are individual objects containing one or more attribute name-value pairs.

`replaceExistingAttributes`	Indicates whether the content to be put will overwrite the existing one, if the item(s) to be put contains any common attribute(s). Applies to the operations `Put` and `BatchPutAttributes` .

Status values

`1`	The operation was executed successfully
`0`	The domain does not exist
`-1`	An error occurred during the operation's execution

Input

The adapter expects an XML document as 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>
```

> In case of the operation `Put` only one item attribute should be provided. If operations for more than one item are to be performed, the operation `BatchPutAttributes` should be used instead. There are no restrictions on element names, however, the attribute `name` is required.

- 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>
```

> In case of the operation `BatchPutAttributes` more than one item attribute should be provided. If operations for only one item are to be performed, the operation `Put` should be used instead. There are no restrictions on element names, however, the attribute `name` is required.

- Operation `Select`:

```
<?xml version="1.0" encoding="UTF-8"
<RootElement>
    <queryStatement>
        select * from 'DomainName'
    </queryStatement>
</RootElement>
```

> The element `queryStatement` is required. For a proper query, follow this guide.

- 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>
```

- The expected input XML structure can be provided using the Amazon S3 Simple Storage Connector's `Search` operation. Moreover, the property `includeObjectsMetadata` hast to be enabled.
- 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.

```
<?xml version='1.0' encoding='UTF-8'?>
<ListBucket>
    <Bucket name="testmalaga2" region="euwest1">
        <Object name="testing_folders/personalData.txt" url="https://
s3euwest1.amazonaws.com/testmalaga2/testing_folders/personalData.txt">
            <Metadata lastModified="2018-12-10 12:30:35.000 +0100"
instanceLength="33" eTag="7960fb01e91834eb9d5ccbacdf45d0e5" contentType="applic
ation/octetstream" contentLength="33" contentLanguage="Spanish"/>
            <UserMetadata ciudad="Malaga"/>
        </Object>
    </Bucket>
</ListBucket>
```

Output

The adapter returns different kind of documents depending on the executed operation:

- 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>
```

- Operation `DomainMetadata` :

```
<?xml version="1.0" encoding="UTF-8"
<Domain domainName="Employee">
    <Metadata attributeNamecount="0"/>
    <Metadata attributeNamesSizeBytes="0"/>
    <Metadata attributeValueCount="0"/>
    <Metadata attributeValueSizeBytes="0"/>
    <Metadata itemCount="0"/>
    <Metadata itemNameSizeBytes="0"/>
    <Metadata timestamp="1544698002"/>
</Domain >
```

- All other operations return the input document.

10.3.3 Dropbox Connector

`Dropbox Connector`: Connects to the Dropbox cloud service and provides cloud storage management functions.

⚠ A Dropbox account (Standard or Business) is required to use the Dropbox Connector.

Before using the adapter for the first time:

If the Dropbox Connector adapter is to be used for the first time, the following steps must be executed beforehand:

1. Go to the website https://www.dropbox.com/developers.
2. Click on **App Console**. The button is located at the top right.
3. Click on **Create app**.
    a. For **Choose an API**, select `Dropbox API`.
    b. For **Choose the type of access you need** choose `Full Dropbox`.
    c. Enter any name for **Name your app**.
    d. Click on **Create app**
    e. The app is created and the app settings open.
4. Klick on **Generate** under **Generate access token.**
5. The Access Token is generated and displayed. This Access Token is required for using the Dropbox Connector.

⚠ Via the API, an `Apps` folder is created in the assigned Dropbox, this forms the application folder. Anyone who is in possession of the generated Access Token has access to this folder! Pay attention to who comes into possession of the Access Token!

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `CopyFile`: Copies a file within the Dropbox. • `CreateFile`: Adds a file to the Dropbox. • `DeleteFile`: Deletes a file within the Dropbox. • `ExtractFolder`: Extracts a .zip file to the folder specified in the parameters. • `GetFileContent`: Outputs the content of a file. • `GetFileMetadata`: Outputs the metadata of a file. • `ListFolder`: Lists the contents of a folder. • `UpdateFile`: Updates a file.

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** *en.softproject.x4.adapter.dropbox.DropBoxConnector*: Main class (default)
`accessToken`	*Required*. Required to connect to the Dropbox account. **Possible values:** Unique Access Token ⓘ The Access Token must be generated in Dropbox.
`folderPath`	Path to a folder in Dropbox. **Possible values:** Folder path (default: `/` (top folder level)) ⚠ The folder path must always begin and end with a `/`. ⓘ If folders are used during the `CreateFile` operation that do not exist, then they will be created automatically.
`fileName`	File name of the file to be processed. **Possible values:** File name incl. file extension
`sourcePath`	*Required for the `CopyFile` operation*. Specifies which file is to be copied. **Possible values:** File path incl. file name and file extension, e.g. `folder/file.txt`

`destinationPath`	*Required for the `CopyFile` operation*. Specifies which file is to be copied. **Possible values:** File path
`folderName`	*Required for the `ExtractFolder` operation*. Name of the folder where the extracted files will be created or overwritten. **Possible values:** Any string
`overwrite`	Defines whether existing files are overwritten when copying or creating. **Possible values:** • `true`: If a file with the same name already exists at this location, then this file will be overwritten. • `false`: If a file with the same name already exists in this location, then the new file will be renamed, e.g. from `File.txt` to `Copy_of_File.txt`.

Status values

`1`	The operation was executed successfully.
`-1`	The operation could not be executed. The error message is output as a text document and in the server log.

Input

The adapter expects a file as input for the operations `CreateFile` and `UpdateFile`.

The adapter expects a .zip file as input for the `ExtractFolder` operation.

Output:

The adapter provides different data as output depending on the executed operation.

- `CopyFile`, `CreateFile` and `DeleteFile` operations: Text document with information
- `ListFolder`, `GetFileMetadata` operations: XML document with corresponding content

Sample output 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>
```

Sample output 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`: Does not return any output.
- Operation `GetFileContent`: content of the read file

> ⚠ Not all file formats are supported!

10.3.4 Google Cloud Storage Objects Connector

`Google Cloud Storage Objects Connector`: Allows to store, search, download and delete objects in google cloud.

Properties

`Operation`	Defines the operation executed by the adapter
	
	Possible values:
	
	• `Search`: Searches for information (buckets or objects) in the Google Cloud Storage
	Required parameters: `bucketName`, `objectName`
	
	• `Upload`: Uploads or creates buckets and objects to the Google Cloud Storage
	Required parameters: `bucketName`, `objectName`, `bucketRegion`, `storageClass`, `fileLocation`, `contentType`
	
	• `Download`: Downloads an object from the Google Cloud Storage
	Required parameters: `bucketName`, `objectName`
	
	• `Delete`: Deletes a bucket or an object in the Google Cloud Storage
	Required parameters: `bucketName`, `objectName`
	
	• `CopyObject`: Copies an Object from a bucket to another one. There are several possibilities to copy objects:
	Required parameters: `bucketName`, `objectName`
	
	  • Copy object within a bucket and change its name.
	Required parameters: `targetObjectName`
	
	  • Copy object into another bucket and change its name.
	Required parameters: `targetObjectName`, `targetBucketName`
	
	  • Copy object into another bucket and remain its name.
	Required parameters: `targetBucketName`
	
	• `MoveOrRenameBucket`. Moves or renames a whole bucket.
	Required parameters: `bucketName`, `targetBucketName`, `targetBucketRegion`
	
	ⓘ • If target bucket name already exists, the output is an error.
	• If any object in target bucket already exists and `overwrite` option is checked, existing objects are overwritten. If `overwrite` option is unchecked, output is an error.
	
	ⓘ Since a bucket can be moved to another geographic region, you must be aware of the limitations associated with moving it to other geographic regions. More information on bucket locations.

- `MoveOrRenameObject`: Moves or renames an object
  Required parameters: `bucketName`, `objectName`, `targetBucketName`, `targetObjectName`

  > If the object already exists in target bucket and `overwrite` option is checked, existing object is overwritten. If `overwrite` option is unchecked, output is an error.

- `UpdateObjectStorageClass`: Changes `storageClass` related with object. Property `storageClass` is closely related with bucket region. For example, when creating a bucket the region must be specified.
  Required parameters: `bucketName`, `objectName`, `storageClass`

  > Not all possible values are allowed for every object/bucket. It depends on bucket specifications and object itself. More information on storage classes.

  > A regional bucket can only store regional objects. In this case, `storageClass` can only be changed in `nearline` or `coldline`.
  >
  > A multi-regional bucket can only store multi-regional objects. In this case `storageClass` can only be changed in `nearline` or `coldline`.
  >
  > **Possible values:**
  >
  > - Object is stored in regional or multi-regional bucket: `coldline`, `nearline`
  > - Object is stored in nearline bucket and bucket region is regional or multi-regional: `coldline`, `regional`. If bucket region is not regional or multi-regional: only `coldline`
  > - Object is stored in coldline bucket and bucket region is regional or multi-regional: `nearline`, `regional`. If bucket region is not regional or multi-regional: only `nearline`

  > If a regional region is specified, e.g. `europe-west1`, it is implicitly specified that default `storageClass` is regional, nearline or coldline, but never multi-regional.
  >
  > Same happens if a multi-regional region is specified, e.g. `us`: It is implicitly specified that default `storageClass` is multi-regional, nearline or coldline, but never regional.

- `AddGrantsPermission`: Adds an ACL permission to an object or a bucket. If the object or bucket already has a ACL permission, nothing happens.

Required parameters:

- For this operation parameters `permissionEntity`, `permissionRole`, `permissionName` and, if set, `PermissionProjectRole` are required.
- Add permission to an object: `objectName` and `bucketName`
- Add permission to a bucket: `bucketName`

If a new permission is added to a bucket, the bucket gets the permission, but not the objects inside. This means that anyone can see a public bucket and the list of its objects, but can't get access to any of the objects. An extra permission is required for this.

- `RevokeGrantsPermission`: Removes an ACL permission from an object or bucket. If the object or bucket has no ACL permission, nothing happens.

Required parameters:

- For this operation parameters `permissionEntity`, `permissionRole`, `permissionName` and, if set, `permissionProjectRole` are required.
- Remove permission from object: `objectName` and `bucketName`
- Remove permission from bucket: `bucketName`

Parameters

`Adapter`	Adapter main class (do not change!) **Possible values:** *de.softproject.integration.adapter.google.cloud.storage.GoogleCloudStorageObjectAdapter*: Main class (default)

`serviceAccountFile`	Path to the JSON file containing the service account key to authenticate to the Google Cloud Storage **Possible values:** • System path, e. g. *`C:\X4\V_5.9\Server\X4DB\1\ExampleProcess\GoogleCloud\Config\serviceAccountFile.json`* • X4db url, e. g. *`x4db://1/ExampleProcess/GoogleCloud/Config/serviceAccountFile.json`* • Xstore url, e. g. *`xstore://ExampleProcess/GoogleCloud/Config/serviceAccountFile.json`*
`bucketName`	Name of the bucket where a set of objects are organized **Possible values:** String containing the name Bucket names must meet the following naming rules: • Bucket names must contain only lowercase letters, numbers, dashes (–), underscores (_), and dots (.). ℹ Names containing dots require verification. • Bucket names must start and end with a number or a letter. • Bucket names must contain 3 to 63 characters. Names containing dots can contain up to 222 characters, but each dot-separated component can't be longer than 63 characters. • Bucket names cannot be represented as an IP address in dotted-decimal notation (e. g. *192.168.5.4*). • Bucket names cannot begin with the `goog` prefix. • Bucket names cannot contain `google` or close misspellings, such as `g00gle`.
`targetBucketName`	*Optional*. Name of the target bucket. For operations that copy/move/rename a bucket. This parameter has the same restrictions as parameter `bucketName`.
`bucketRegion`	Region where the buckets are defined **Possible values:** String containing the region *Example*: • Regional location: `europe-west3` (Europe, Frankfurt) • Multi-regional location: `eu` (data centers in the European Union)

`targetBucketRegion`	Name of the target bucket's region. For operations that move/rename a bucket and where a target region can be specified. This parameter has the same restrictions as parameter `bucketRegion`.
`storageClass`	Storage class of the bucket and the objects. This parameter is closely related with regions, since a storageClass value can only be assigned to a bucket or object that is related with the choosen `bucketRegion` parameter value. **Possible values:** Combo with following values • `multi_regional` • `regional` • `nearline` • `coldline`
`objectName`	Name of the object. Required for operations `AddGrantsPermission` and `RevokeGrantsPermission`. **Possible values:** String containing the name Object names must meet the following naming rules: • Object names can contain any sequence of valid Unicode characters with a length between 1-1024 bytes, if they are UTF-8 encoded. • Object names cannot contain carriage return or line feed characters. • Object names cannot start with `.well-known/acme-challenge`. • Objects cannot be named `.` or `..` *Examples*: • *`document`* • *`folder/subfolder/document`* • *`document.txt`*
`targetObjectName`	For operations that copy/move/rename an object. This parameter has the same restrictions as parameter `objectName`.
`fileLocation`	Path of the file to be uploaded to the Google Cloud Account; This parameter is only valid for the `Upload` operation **Possible values:** • System path, e. g. *`C:\X4\V_5.9\Server\X4DB\1\ExampleProcess\GoogleCloudStorageObject/Data/TextDocument.txt`* • X4db path, e. g. *`x4db://1/ExampleProcess/GoogleCloudStorageObject/Data/TextDocument.txt`* • Xstore path, e. g. *`xstore://ExampleProcess/GoogleCloudStorageObject/Data/TextDocument.txt`*

`contentType`	Content type of the object to be uploaded **Possible values:** String containing a valid mime-type for the object to be uploaded, e. g. *`text/plain`*, *`application/xml`* etc.
`overwrite`	Indicates if during an operation that moves/copies/renames a bucket or object, buckets or objects with the same name are overwritten. If this option is unchecked and objects/buckets with the same name exist at the target location, the output contains an error. **Possible values:** • *`checked`*: Existing objects/buckets are overwritten. • *`unchecked`*: Existing objects/buckets are not overwritten, output contains an error.
`permissionEntity`	Type of ACL permission that is created. **Possible values:** Combo with following values • `Project`: Specifies a permission based on a list of project members. Parameter `permissionProjectRole` must be provided to specify from which sublist the members should be selected. • `User`: Specifies a permission based on a specific email address, a "public access" permission or access for all Google accounts. • `Domain`: Specifies a permission based on a internet domain name thas is associated with G Suite or Cloud Identity. • `Group`: Specifies a permission based on a group of Google users. Uses a specific email address that is related to a Google group.
`permissionRole`	Specifies if the permission grants read or write access. **Possible values:** Combo with following values • `READER`: Read access for object/bucket • `WRITER`: Read and write access for object/bucket • `OWNER`: Managing of object/bucket. Contains access rights of `WRITER`.

`permissionName`	Specifies a specific value that is used in every type of permission.
	
	Possible values: depending on the value of `permissionEntity`
	
	• `permissionEntity="Project"`: Project number. Creates a permission for `OWNERS`, `EDITORS` and `VIEWERS` that are related with this project.
	• `permissionEntity="User"`:
	  • `allUsers`: Creates a public permission.
	  • `allAuthenticatedUsers`: Creates a permisson for all Google accounts.
	  • `<email address>`: Creates a permission for the user registered with this email address.
	• `permissionEntity="Domain"`: `<USERNAME>@<YOUR_DOMAIN>.com`: Creates a permission for owners of a email address that is related to a specific internet domain name.
	• `permissionEntity="Group"`: Email address of a Google group. Creates a permission for all members of this Google group. More information on Google groups.

`permissionProjectRole`	Parameter affects parameters `permissionName` and `permissionEntity` if `permissionEntity="Project"`. **Possible values:** Combo with following values • `OWNERS` • `EDITORS` • `VIEWERS`

Status values

`1`	The adapter operation was executed successfully
`-1`	An error occurred during the adapter's execution
`0`	• Operation `Upload`: the object to be uploaded already exists • Operation `Download`: the object to be downloaded doesn't exist • Operation `Delete`: the object or bucket to be deleted doesn't exists

Input

For some operations the adapter expects any input document (binary document or a base64 document).

- Operation `Upload` : If the parameter `fileLocation` is specified, an input document is not required. Otherwise an input is required.
- All other operations don't require an input document.

Output

The adapter returns different kind of documents depending on the executed operation:

- Operation `Upload` :
  - If the adapter operation was executed successfully (status `1`), the adapter returns the input document.
  - If an error occurred during the adapter's execution (status `-1`), the output will be a new document with the error details.
  - If the object to be uploaded already exists (status `0`), the output corresponds to the input document.
- Operation `Search`:
  - If the adapter operation was executed successfully (status `1`), the adapter returns an XML document with all buckets and theirs objects:

Sample output for the 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>
```

  - If an error occurred during the adapter's execution (status `-1`), the output will be a new document with the error details.
- Operation `Download`:
  - If the adapter operation was executed successfully (status `1`), the output document is the downloaded object (text, XML, binary doc etc.).

    - If an error occurred during the adapter's execution (status `-1`), the output will be a new document with the error details.
    - If the blob or container don't exist (status `0`), the output corresponds to the input document.
- Operation `Delete`:
    - If the adapter operation was executed successfully (status `1`), the output corresponds to the input document.
    - If an error occurred during the adapter's execution (status `-1`), the output corresponds to the input document.
    - If the bucket or object to be deleted don't exist (status `0`), the output corresponds to the input document.
- Operation `CopyObject`, `MoveOrRenameBucket`, `MoveOrRenameObject`, `UpdateObjectStorageClass`:
    - No output for regular execution. If the operation cannot be executed, the output contains an error.

10.3.5 Google Drive Connector

`Google Drive Connector`: Connects to the Google Drive cloud service and provides cloud storage data management capabilities.

> ⚠ Google Drive allows to have multiple objects with the same name. Currently, when executing operations that refer to multiple existing objects, an error message is issued and the operation is not executed.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `CopyFile`: Copies an existing file within the Google Drive account • `CreateComment`: Creates a comment • `CreateFolder`: Creates a new folder in the Google Drive account • `DeleteFolder`: Deletes an already existing folder • `DeleteFile`: Deletes an already existing file • `ExtractArchiveToFolder`: Extracts files inside a ZIP folder to a folder • `GetFileContent`: Retrieves the contents of an already existing document • `GetFileMetadata`: Gets the metadata of a file • `ListFiles`: Lists all files and folders of the root directory • `ListFilesInFolder`: Lists all files and folders inside a specified folder • `ReadComments`: Reads out already existing comments • `RevokePermission`: Revokes permission according to the parameter configuration. • `SetPermission`: Creates or updates permissions according to the parameter configuration. • `UpdateFile`: Updates the content or the name of a file • `UpdateFolderName`: Updates the name of an already existing folder • `UploadFile`: Uploads a file to the Google Drive account.

Parameters

`Adapter`	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.google.cloud.drive.GoogleCloudDriveAdapter*: Main class (default)
`serviceAccountFile`	Path of the service account file for authentication **Possible values:** String with the path of the service account file.
`folderPath`	Folder path where the extracted file will be placed • If the path is not specified, the path of the root directory is used. • This parameter is used for `ListFilesInFolder`, `DeleteFile`, `GetFileContent` and `GetFileMetadata` operations. **Possible values:** String with the folder path; The path specification must end with a slash `/`.

`sourcePath`	Path of the source file ⓘ This parameter is used for `UploadFile`, `UpdateFile`, `CopyFile` and `ExtractArchiveToFolder` operations. **Possible values:** string with the path to the source file
`destinationPath`	Path to the destination folder or file ⓘ This parameter is used for `CopyFile`, `UploadFile`, `UpdateFile` and `ExtractArchiveToFolder` operations. **Possible values:** String with the path to the destination file or folder; The path specification must end with a slash `/`.
`filename`	Filename - in some cases the parameter is used to set or change the filename. ⓘ This parameter is used for `UploadFile`, `DeleteFile`, `GetFileContent`, `GetFileMetadata`, `UpdateFile` and `CopyFile` operations. **Possible values:** String with the file name
`folderName`	Folder name ⓘ • For the `ExtractArchiveToFolder` operation, the parameter is used to create a new folder where the files are extracted from the ZIP folder. • The parameter is also used for the `CreateFolder`, `UpdateFolderName` and `DeleteFolder`operations. **Possible values:** String with the folder name
`comment`	Comment ⓘ This parameter is used for *create* and *read* operations. **Possible values:** String with the comment

`overwrite`	Specify whether existing files are to be overwritten ⓘ This parameter is used for all *create*, *copy* or *update* operations except for the `CreateFolder` operation. For *`false`* a message is output if the file/folder already exists. **Possible values:** Boolean value *`true`* or *`false`*.
`scope`	Scope of the authorization ⓘ This parameter is used for the `SetPermission` operation. **Possible values:** • `DOMAIN` • `GROUP` • `USER` • `ANYONE`
`role`	Role of authorization ⓘ This parameter is used for the `SetPermission` operation. **Possible values:** • `FILE_ORGANIZER` • `READER` • `OWNER` • `WRITER` • `COMMENTER`
`account`	Domain or e-mail address of the user or group ⓘ This parameter is used for `SetPermission` and `RevokePermission` operations. **Possible values:** String with the domain or e-mail address

Status values

`1`	The adapter operation was executed successfully
`-1`	An error occurred while executing the adapter

Input

The adapter is managed via parameters. However, for some operations such as `ExtractArchiveToFolder`, `UploadFile` or `UpdateFile`, an input can be passed that represents the

content to be uploaded or updated. If no input is passed for these operations, the `sourcePath` parameter is used.

Output

The adapter outputs the following structure when a *list* operation is used:

Output for list operations

```
<?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>
```

The following information is output for the `GetFileMetadata` operation:

Output for the GetFileMetadata operation

```
<?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`: Enables the connection to a Microsoft Azure Blob Storage.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** ⓘ The parameters `containerName` and `blobname` are required for all operations. • `Search`: Searches information (containers or blobs) in the Microsoft Azure storage account • `Append`: Appends information to the selected blob in the Microsoft Azure storage account **Required parameters:** `fileLocation` • `Upload`: Uploads or creates containers and blobs to the Microsoft Azure storage account **Required parameters:** `fileLocation` • `Download`: Downloads a blob from the Microsoft Azure storage account • `Delete`: Deletes a container or a blob in the Microsoft Azure storage account • `UpdateDirectory`: Updates the source directory name with the target directory name **Required parameters:** `sourceDirectoryName`, `targetDirectoryName` • `MoveDirectory`: Moves the content from the source directory to the target directory **Required parameters:** `sourceDirectoryName`, `targetDirectoryName` • `DeleteDirectory`: Deletes the content from the source directory **Required parameters:** `sourceDirectoryName`, `targetDirectoryName` • `MoveBlobDirectory`: Moves a specific blob from the source directory to the target directory **Required parameters:** `sourceDirectoryName`, `targetDirectoryName`

Parameters

`Adapter`	Adapter main class (do not change!) **Possible values:** *de.softproject.integration.adapter.microsoft.azure.blob.MicrosoftAzureBlobStorageAdapter*: main class (default)
`accountName`	Name of the Microsoft Azure storage account **Possible values:** Any valid string, e. g. *`softproject`*
`accountKey`	Password of the Microsoft Azure storage account **Possible values:** Any string containing the key, e. g. *`AD9ohFhJ6Hve9ODJEXTw53MjR1tjoUZ8lWfDlw5UFqGMih3VtUCapIwIfEMBmITeShQN9MbCITE4SzxTcGIybw==`*
`containerName`	Name of the container where the blobs are organized **Possible values:** Any string containing the name The container name must meet the following naming rules: • Container names must start with a letter or number, and can contain only letters, numbers, and the dash (–) character • Every dash (–) character must be immediately preceded and followed by a letter or number; consecutive dashes are not permitted in container names • All letters in a container name must be lowercase • Container names must be between 3 and 63 characters long
`blobName`	Name of the blobs stored **Possible values:** Any string containing the name The blob name must meet the following naming rules: • A blob name can contain any combination of characters • A blob name must be at least one character long and cannot be more than 1,024 characters long • Blob names are case-sensitive • Reserved URL characters must be properly escaped • The number of path segments comprising the blob name cannot exceed 254. A path segment is the string between consecutive delimiter characters (e. g., the forward slash '/') that corresponds to the name of a virtual directory

`fileLocation`	Path of the file to be uploaded to the storage account; This parameter is only valid for the operation `Upload` **Possible values:** • System path, e. g. *`C:\X4\V_5.9\Server\X4DB\1\ExampleProcess/MAzureBlobStore/Data/TextDocument.txt`* • X4db path, e. g. *`x4db://1/ExampleProcess/MAzureBlobStore/Data/TextDocument.txt`* • Xstore path, e. g. *`xstore://ExampleProcess/MAzureBlobStore/Data/TextDocument.txt`*
`overwrite`	Flag indicating whether blobs are overwritten or not
`sourceDirectoryName`	Name of the source directory for all directory operations **Possible values:** Any valid string ending with `/`, e. g. *`sourceFolder/`*
`targetDirectoryName`	Name of the target directory for all directory operations **Possible values:** Any valid string ending with `/`, e. g. *`targetFolder/`*

Status values

`1`	The operation was executed successfully
`-1`	An error occurred during execution

`0`	• Operation `Download`: the blob to be downloaded doesn't exist
	• Operation `Delete`: the blob to be deleted doesn't exist
	• Operations `UpdateDirectory`, `MoveDirectory`, `DeleteDirectory`, `MoveBlobDirectory`:
	  • the source directory doesn't exist
	  • the target directory could not be found (operations `MoveDirectory`, `MoveBlobDirectory`)
	  • the blob doesn't exist in the source folder (operation `MoveBlobDirectory`)

Input

The adapter expects any input document, e. g. a binary document or a base64 document (XML or text).

- Operation `Upload`: An input document can be omitted, if the parameter `fileLocation` is specified. In any other cases an input document is required.
- For all other operations, an input document is not required.

Output

The adapter returns different kind of documents depending on the executed operation:

Operation `Upload`:

- If the operation was executed successfully (status `1`), the adapter returns an XML document with the URL of the uploaded file or document.

  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>
  ```

- In case of an error (status `-1`), the output corresponds to the input document.

Operation `Search`:

- If the operation was executed successfully (status `1`), the adapter returns an XML document with all containers and their blob URI objects.

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>
```

- In case of an error (status `-1`), the output corresponds to the input document.

Operation `Download`:

- If the operation was executed successfully (status `1`), the adapter returns the downloaded blob (text, XML, binary document etc.).
- In case of an error (status `-1`), the output corresponds to the input document.
- If the blob or container don't exist (status `0`), the output document corresponds to the input document.

Operation *`UpdateDirectory`*:

- If the operation was executed successfully (status `1`), the output document shows all affected blobs after updating the source folder with the name of the target folder.

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>
```

- In case of an error (status `-1`), the output corresponds to the input document.
- If the source directory doesn't exists (status `0`), the output is an empty XML document.

Operation UpdateDirectory: Status 0

```
<?xml version="1.0" encoding="UTF-8"
<UpdateDirectory targetDirectory="otherFiles/testSubFolderR/" sourceDirectory="
textFiles/subTextFiles"/>
```

Operation `MoveDirectory`:

- If the operation was executed successfully (status `1`), the output document shows all affected blobs after moving the source folder to the target folder.

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>
```

- In case of an error (status `-1`), the output corresponds to the input document.
- If the source or target directory doesn't exists (status `0`), the output is an empty XML document.

Operation MoveDirectory: Status 0 - Target directory doesn't exist

```
<?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 - Source directory doesn't exist

```
<?xml version="1.0" encoding="UTF-8"
<MoveBlobDirectory targetDirectory="textFiles/" sourceDirectory="otherFiles/
testSubFolderR"/>
```

Operation `DeleteDirectory`:

- If the operation was executed successfully (status `1`), the output document shows all affected blobs after deleting the source folder and its content

Operation DeleteDirectory: Status 1

```
<?xml version="1.0" encoding="UTF-8"
<DeleteDirectory targetDirectory="" sourceDirectory="imagesFiles2/">
    <Blob newName="" name="imagesFiles2/image1.png"/>
</DeleteDirectory>
```

- In case of an error (status `-1`), the output corresponds to the input document.
- If the source directory doesn't exists (status `0`), the output is an empty XML document.

Operation DeleteDirectory: Status 0

```
<?xml version="1.0" encoding="UTF-8"
<DeleteDirectory targetDirectory="" sourceDirectory="imagesFiles2/"/>
```

Operation `MoveBlobDirectory`:

- If the operation was executed successfully (status `1`), the output document shows the blobs moved from the source to the target folder.

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>
```

- In case of an error (status `-1`), the output corresponds to the input document.
- If the source or target directory doesn't exists (status `0`), the output is an empty XML document.

Operation MoveBlobDirectory: Status 0 - Target directory doesn't exist

```
<?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 - Source directory doesn't exist

```
<?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`: Connects to Microsoft Azure, a cloud computing platform from Microsoft that provides Cloud Cosmos DB Table Storage.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `CreateTable`: Creates a new table. **Required parameters:** `tableName` • `ListTable`: Lists all tables. • `DeleteTable`: Deletes an already existing table. **Required parameters:** `tableName` • `InsertEntity`: Inserts the entity, if the entity does not exist, or replaces an existing entity. Once the operation was executed successfully, the table will contain the new entity with properties as defined in the request and the prior entity and its properties will be replaced. **Required parameters:** `tableName`, `partitionKey`, `rowKey` • `InsertBatchEntity`: Inserts a batch of entities in one write operation using `InsertOrReplace` entity. **Required parameters:** `tableName` • `RetrieveEntity`: Retrieves a single, specified entity. **Required parameters:** `tableName`, `partitionKey`, `rowKey` • `RetrieveAllEntities`: Retrieves all entities from a specific `partitionKey`. **Required parameters:** `tableName`, `partitionKey` • `RetrieveRangeEntities`: Retrieves a range of entities using the `partitionKey` and filters on a range of `rowKey` values. `partitionKey` identifies a specific partition, `rowKey` values identify a subset of the entities in that partition. **Required parameters:** `tableName`, `partitionKey`, `rowKey`, `filterCondition` • `DeleteEntity`: Deletes a single entity. **Required parameters:** `tableName`, `partitionKey`, `rowKey` • `ModifyEntity`: Modifies an entity, choosing between a merge or replace operation using `replaceExistingItems` parameter check. If replace operation ist selected (`replaceExistingItems="true"`), only the provided properties will remain after the operation, the other properties will be deleted. If merge operation is selected (`replaceExistingItems="false"`), all properties will remain and the provided ones will be updated. **Required parameters:** `tableName`, `partitionKey`, `rowKey`, `replaceExistingItems`

	• `QuerySubsetOfEntityProperties`: Gets a subset of entity properties according to an XML input. If `rowKey` and `partitionKey` are specified, only matching entity properties will be retrieved. If `rowKey` and `partitionKey` are not specified, all entities in that table will be retrieved. The operation's output is an XML file containing the subset of properties inside a parent element for each related entity. **Required parameters:** `tableName`, `partitionKey`, `rowKey`

Parameters

`Adapter`	Adapter main class (do not change!) **Possible values:** *de.softproject.integration.adapter.microsoft.azure.table.MicrosoftAzureTableStorageAdapter*: main class (default)
`accountName`	Name of the Microsoft Azure storage account **Possible values:** Any valid string, e.g. `softproject`
`accountKey`	Password of the Microsoft Azure storage account **Possible values:** Any string containing the serial key provide by Microsoft Azure, e. g. *AD9ohFhJ6Hve9ODJEXTw53MjR1tjoUZ8lWfDlw5UFqGMih3VtUCapIwIfEMBmITeShQN9MbCITE4SzxTcGIybw==*
`endPoint`	Azure Cosmos DB endpoint
`tableName`	Name of the table where entities are stored **Possible values:** String containing the ID for the new table Table names have the same character requirements as database IDs. Database names must be between 1 and 255 characters, and cannot contain `/ \ # ?` or a trailing space.
`partitionKey`	Element of the primary key Entities with the same partition key can be queried faster, and inserted/updated in atomic operations. **Possible values:** Any string containing the partition key
`rowKey`	Element of the primary key An entity's row key is its unique identifier within a partition **Possible values:** Any string containing the row key

`filterCondition`	Use to compare `partitionKey` an `rowKey` **Possible values:** • EQUAL • GREATER_THAN • GREATER_THAN_OR_EQUAL • LESS_THAN • LESS_THAN_OR_EQUAL • NOT_EQUAL
`replaceExistingItems`	Indicates if `ModifyEntity` performs a replace or a merge operation. For further information please check http://azure.github.io/azure-storage-java/. **Possible Values:** • Checked: perform replace operation • Unchecked: perform merge operation

Status values

`1`	The operation was executed successfully
`-1`	An error occurred during the operation's execution
`0`	Operation `CreateTable`: The table could not be created, but neither an error nor an exception were returned. Operation `DeleteTable`: The specified `tableName` does not exist.

Input

The adapter expects an XML document as input

- In case of `InsertEntity` operation, an xml with the following structure is expected:

Expected structure for 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>
```

ⓘ The property (key) `name` cannot be protected words like *`Id`*.

- In case of `ModifyEntity` operation, an xml with the following structure is expected:

Expected structure for operation ModifyEntity

```
<?xml version="1.0" encoding="UTF-8" ?>
<Entity>
    <Properties name="Name" value="Carlos" />
</Entity>
```

- In case of `QueryOfSubsetEntityProperties` operation, an xml where the required properties are specified is expected:

Expected structure for operation QueryOfSubsetEntityProperties

```
<?xml version="1.0" encoding="UTF-8" ?>
<Entity>
    <Property name="Department" />
    <Property name="Surname" />
    <Property name="nonExistingProperty" />
</Entity>
```

Output

The adapter returns different kind of documents depending on the executed operation:

- Output for operation `RetrieveEntity`:

Output for 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="Name" value="Paolo"/>
    <Properties name="Surname" value="Green"/>
</Entity>
```

- Output for operations `RetrieveAllEntities` and `RetrieveRangeEntities`:

Output for operations RetrieveAllEntities and 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>
```

- Output for operation `QueryOfSubsetEntityProperties`:

Output for 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>
```

ⓘ Properties `partitionKey` and `rowKey` are always returned because they are used to uniquely identify the entity. Since the operation is able to return multiple properties of more than one object, it is important to assign the properties to the associated entity.

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`: Calls programmes on *AS/400* systems (*IBM eServer iSeries* or *System i*) via the *Program Call Markup Language* (PCML) and returns the results of the executed programme as an XML document.

Former designation up to version 4.5.1: *AS/400 PCML Connection Adapter*

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Execute`*: Execute adapter

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.PCMLAdapter`*: Main class (default)
systemName	Host name of the AS/400 system **Possible values:** Any host name (e.g. *`localhost`*)
userId	Username for server authentication **Possible values:** • Any string • *`*CURRENT`*: call up current user ID
password	Password for server authentication **Possible values:** Any string
encoding	Character encoding of the input and output XML document **Possible values:** Any valid character encoding (e.g. *`UTF-8`*)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

This adapter expects an adapter-specific input XML structure. Each `<program>` element corresponds to a programme, each `<data>` element therein defines a field, and each `<structure>` element can serve as a data structure element or an argument for a programme:

```
<pcml version="1.0">
   <program>
      <data></data>
      <struct>
          <data></data>
      </struct>
   </program>
</pcml>
```

Output

The successfully executed PCML adapter returns an XML document with the following pattern:

```
<pcml>
   <program_name>
      <data_values></data_values>
      <struct_name1></struct_name1>
   </program_name>
</pcml>
```

10.4.2 File System

`File System`: Stores, reads, deletes, copies, moves, lists files and folders or attaches files to a target file located in the file system of the X4 server or on connected Windows or Samba network drives.

> ⚠ The *File System* adapter must not be used to modify files in the X4 repository, since it does not access the X4 repository but the file system directly. If the adapter is used for this purpose, error messages and malfunctions may occur.
> To modify files in the X4 repository, use the adapter Projects or Project Access.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Write File`: Write data from the previous process step to the target file defined in the `path` parameter. • `Append File`: Append file from the previous process step to the target file defined in the `path` parameter. ⓘ If the target file defined in the `path` parameter does not exist, a file will be created. • `Read File`: Read file defined in the `path` parameter. • `Delete`: Delete files or folders defined in the `path` parameter. • `Copy`: Copy files or folders defined in the `path` parameter to the folder defined in the `targetPath` parameter. • `Move`: Move files or folders defined in the `path` parameter to the folder defined in the `targetPath` parameter. • `Exist`: Check whether the files or folders defined in the `path` parameter exist. Outputs an XML with metadata as a result. • `Create Folder`: Creates the folder defined in the `path` parameter. • `List Folder`: Lists names and metadata of files and folders from the folder defined in the `path` parameter. Outputs an XML with metadata as a result. ⓘ The content of the files is not listed.

Parameters

`Adapter`	Adapter main class (do not change!) **Possible values:** • `de.softproject.integration.adapter.filesystem.FileSystemAdapter`: Main class (default)
`path`	Path to a file or folder in the file system of the X4 server or to a connected Windows or Samba network drive. To do this, the X4 server requires appropriate permissions on the destination folder. **Possible values:** • Any string, if necessary including placeholders and file extension e.g. *`%USER_COLLECTION%/Data/Input.xml`* • *`c:/`*: Access the W Windowssystem partition C:\. • Any UNC network drive specification according to the scheme `\\\\Server\\share\\public\folder\\\` with any spaces and special characters e.g. `\\\\SP-PM-01\\share\\Public Folder\\`

`targetPath`	Additional path to a file or folder in the file system of the X4 server or to a connected Windows or Samba network drive. To do this, the X4 server requires appropriate permissions on the destination folder. ⚠ This parameter is used only for `Move` and `Copy` operation. **Possible values:** • Any string, if necessary including placeholders and file extension e.g. *`%USER_COLLECTION%\Data\Input.xml`* • *`c:\`* Access the Windows system partition C:\. • Any UNC network drive specification according to the scheme *`\\\\Server\\share\\public\folder\\\`* with any spaces and special characters e.g. *`\\\\SP-PM-01\\share\\Public Folder\\`*
`nameFilter`	Filters the output of the `ListFolder` operation by name **Possible values:** • Any string (e.g. *`myFolder`*) • Wildcard `*`
`addNewline`	Inserts a new line in the destination file on top of the file to be appended. Can be used only with the `Append File` operation. **Possible values:** • `None`: (default) Does not insert a new line. • `Auto`: Inserts a new line if no new line was inserted before. Detects the format of an already inserted new line. • `Windows`: Inserts a new line in Windows format. • `Unix`: Inserts a new line in Unix format.

Status values

`1`	The adapter operation was executed successfully.
`-1`	The operation failed due to a technical error.

For further information please visit the section

- 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

The `Write File` operation writes data from the previous process step to the destination file defined in parameter `path`.

Examples

The following example shows a sample parameterization of the adapter for the `Write File` operation to write the data from the previous process step to the destination file defined in parameter `path`.

Parameters	Value
`path`	C:\MyFolder\File.xml

For further information please visit the section

- 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

The `Append File` operation appends the data from the previous process step to the destination file specified in parameter `path`.

> If the target file defined in the `path` parameter does not exist, a file will be created.

Examples

The following example shows a sample parameterization of the adapter for the `Append File` operation to append the data from the previous process step to the destination file defined in `path` parameter.

Parameters	Value
`path`	C:\MyFolder\File.xml
`addNewline`	`Auto`

Target file

```
<?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>
```

⚠ The output is not a valid XML file. The output must be transformed into a valid XML file using a mapping.

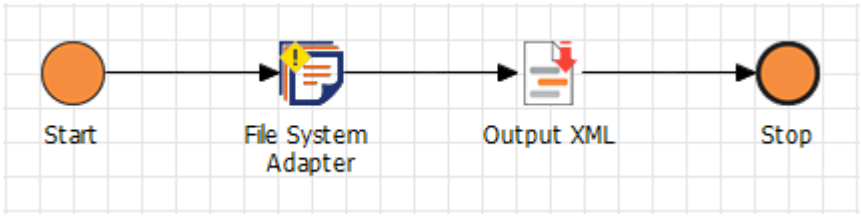
For further information please visit the section

- 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

The `Read File` operation reads the data from the file defined in the `path` parameter.

Examples



The following example shows a sample parameterization of the adapter for the `Read File` operation to read the data from the destination file defined in the `path` parameter.

Parameters	Value
`path`	C:\MyFolder\File.xml

The output of the adapter is then written to an XML file in the X4 repository.

For further information please visit the section

- 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

The `Delete` operation deletes the local file or folder defined in the `path` parameter.

> ⚠ The data will be permanently deleted.

Examples

Delete file

The following example shows a sample parameterization of the adapter for the `Delete` operation to delete the target file defined in the `path` parameter.

Parameters	Value
`path`	C:\MyFolder\File.xml

Delete folder

The following example shows a sample parameterization of the adapter for the `Delete` operation to delete the destination folder defined in the `path` parameter.

Parameters	Value
`path`	C:\MyFolder

> ✓ For further information please visit the section
>
> - 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

The `Copy` operation copies the file or folder defined in the `path` parameter to the local folder defined in the `targetPath` parameter.

Examples

Copy folder

The following example shows a sample parameterization of the adapter for the `Copy` operation to copy the `MyFolder` folder to the `X4` directory:

Parameters	Value
`path`	C:\MyFolder
`targetPath`	C:\X4\MyFolder

Copy file

The following example shows a sample parameterization of the adapter for the `Copy` operation to copy the `MyFile.xml` file to the `X4` directory:

Parameters	Value
`path`	C:\MyFolder\MyFile.xml
`targetPath`	C:\X4\MyFolder\MyFile.xml

For further information please visit the section

- 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

The `Move` operation moves the file or folder defined in the `path` parameter to the local folder defined in the `targetPath` parameter.

Examples

Move folder

The following example shows a sample parameterization of the adapter for the `Move` operation to move the `MyFolder` folder to the `X4` folder:

Parameters	Value
path	C:\MyFolder
targetPath	C:\X4\MyFolder

Move file

The following example shows a sample parameterization of the adapter for the Move operation to move the MyFile.xml file to the0 X4 directory:

Parameters	Value
path	C:\MyFolder\MyFile.xml
targetPath	C:\X4\MyFolder\MyFile.xml

For further information please visit the section

- 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

The Exists operation checks whether the files or folders defined in the path parameter exist. Outputs an XML with metadata as a result.

Status values

1	The adapter operation was executed successfully.
-1	The file or folder does not exist.

Examples

The following example shows a sample parameterization of the adapter for the Exist operation to check whether the target file defined in path parameter exists.

Parameters	Value
path	C:\MyFolder\File.xml

Output

If the file does not exist:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>java.nio.file.NoSuchFileException: C:\MyFolder\File.xml</Error>
```

If the file exists:

```
<?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>
```

ⓘ When checking if a folder exists, the attribute `isDirectory="true"` is set.

For further information please visit the section

- 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

The `Create Folder` operation creates the folder defined in the `path` parameter.

Examples

The following example shows a sample parameterization of the adapter for the `Create Folder` operation to create the folder defined in the `path` parameter.

Parameters	Value
`path`	C:\MyFolder

For further information please visit the section

- 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

The `List Folder` operation lists names and metadata of files and folders from the folder defined in the `path` parameter. Outputs an XML with metadata as a result.

`1`	The adapter operation was executed successfully.
`-1`	The folder doesn't exist.

Examples

The following example shows a sample parameterization of the adapter for the `List Folder` operation to list PDF files from the folder defined in the `path` parameter.

Parameters	Value
`path`	C:\MyFolder
`nameFilter`	*.pdf

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>
```

When a folder is listed, the attribute `isDirectory="true"` is set.

For further information please visit the section

- 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`: Simplifies the use of file operations in Technical Processes. It allows to easily perform actions such as listing, moving, and copying multiple files or directories. The compatibility with the ZIP in Memory Compression (Deprecated) adapter allows to easily (un)pack multiple files and directories.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Read`: Read files • `Write`: Create/replace files and directories • `Delete`: Delete files and directories • `GetMetadata`: Read metadata from files and directories • `Copy`: Copy files and directories • `Move`: Move files and directories • `List`: List files and directories • `ListAndRead`: List files and directories and read the file contents at the same time (combines `List` with `Read`). • `ListWithMetadata`: List files and directories and read the metadata at the same time (combines `List` with `GetMetadata`)

Input and output structures of the *Filesystem Bulk* adapter are compatible with the ZIP in Memory Compression (Deprecated) adapter's structures.

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.filesystem.Bulk:* Main class (Default)
baseDirectory	Base directory for all operations. All operations work exclusively in this directory and its sub-directories. **Possible values:** Absolute path to the base directory
subDirectory	Directory relative to the `baseDirectory` as basis for all operations. **Possible values:** • *<empty>:* operations start within `baseDirectory` • `.`: operations start within `baseDirectory` • *a*: operations start within `<baseDirectory>/a` • `..` or *a/../..*: Error, the `baseDirectory` can not be left this way

limit	Limits the amount of results returned by a list operation. **Possible values:** • *<positiveNumeber>:* maximum number of returned results • *0*: result is an empty list • *<negativeNumber>*: maximum number of results is unlimited ⓘ When searching for files, it is recommended to set the parameter `skipDirectories` to *true*, otherwise the desired matches for directories could be displaced.
maxLevel	Specifies the recursion depth for list operations. **Possible values:** • *<positiveNumber>*: maximum depth a list operation is descending into directories • *1*: only the direct content of the directory, which is specified by `baseDirectory/subDirectory` • *0*: result is an empty list • *<negativeNumber>*: the recursion depth is unlimited
filter	Regular expression applied to the relative path of an entry in case of a list operation. If the regular expression applies, the path will be included in the result set of the list.
skipDirectories	Defines whether a list operation can return directory elements in the result. **Possible values:** • *true*: directories won't show up in the result of list operations (helpful in combination with `limit`) • *false*: directories are included in the result of list operations
overwrite	Defines whether a `Write`, `Move` or `Copy` operation is allowed to overwrite existing files. **Possible values:** • *true:* existing files may be overwritten with `Write`, `Copy` and `Move` • *false:* existing files mustn't be overwritten

strict	Defines whether parent directories have to exist for `write` operations, and a directory has to be empty in case of `delete` operations.
	
	Possible values:
	
	• *true*:
	  • The target directory to which is copied must exist for the operations `Copy` and `Move`.
	  • The directories mus be empty for the operation `Delete`.
	  • The target directory to which is written must exist for the operation `Write`.
	• *false*: The parent directories will be created for `write` operations. Moreover, directories will be deleted even if they are not yet empty.

Status values

1	All entries for the operation could be executed successfully.
0	Not all entries for the operation could be executed successfully, i.e. the result contains `Error` elements.
-1	Other technical problems occurred during the execution, e.g. invalid input.

Input

Input structure for Read, Write, GetMetadata and Delete

```
<any>
 <!-- any mixture of file and directory elements -->
 <File name="relativePath">...Base64 content for Write...</File>
 <Directory name="relativePath" />
</any>
```

Input structure for Copy and Move

```
<any>
 <!-- any mixture of file and firectory 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 and Move only have Error elements as result, if there have actually been errors.

Example

Within the demo system *X4 Experience* you can find the sample process `FilesystemBulk.wrf` which shows the adapter's use in practice.

10.4.3.1 Example for List

Prerequisites

The following directory structure is given:

```
/
+ a/
| + index.html
| + logo.gif
| + content/
|   + toc.html
|   + page1.html
|   + page2.html
|   + images/
|     + pic1.png
|     + pic2.png
+ b/
  + some.html
  + other.html
  + files.html
```

Szenarios

List everything

Adapter configuration	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>

The first five entries

Adapter configuration	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>

The first five files

Adapter configuration	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>

Only the current directory

Adapter configuration	Output
• baseDirectory = /a • subDirectory = • limit = -1 • maxLevel = 1 • filter = • skipDirectories = false	<Data> <File name="index.html" ... /> <File name="logo.gif" ... /> <Directory name="content" ... /> </Data>

All files and directories complying a filter

Adapter configuration	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>

All files complying a filter (without directories)

Adapter configuration	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>

Everything beneath a subdirectory

Adapter configuration	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>

Everything beneath the current directory and the direct child directories

Adapter configuration	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>

The start directory can be located at any depth within the directory

Adapter configuration	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`: Supports FTP (*File Transfer Protocol*), SFTP (*SSH File Transfer Protocol*) and FTPS (*File Transfer Protocol Secure / FTP-SSL*) for sending, receiving, deleting, moving a file or folder and for listing folder contents. The adapter supports proxy servers of type `HTTP`, `SOCKS4` und `SOCKS5`.

Former name up to version 4.5.1: *File Transfer Adapter*

> This adapter replaces the *FTP Transfer* as well as the *SFTP Communication* Adapter.

Properties

`Operation`	Determines which file transfer operation is performed on the processed files or folders. **Possible values:** • `Exists`: Check whether the files or folders specified in the `path` parameter exist. Outputs an XML with metadata as a result. • `List`: List names and metadata of files and folders from the folder specified in the `path` parameter. The output is an XML with metadata as a result. > The content of the files is not listed. • `Send File`: Write data from the previous process step to the destination file specified in the `path` parameter. • `Receive File`: Read the file specified in the `path` parameter. • `Append File`: Append file from the previous process step to the destination file specified in the `path` parameter. • `Move`: Move file or folder on FTP(S)/SFTP server from `path` to `targetPath`. • `Delete File`: Delete files specified in the `path` parameter. • `Create Folder`: Create folder on FTP(S)/SFTP server specified in the `path` parameter. • `Delete Folder`: Delete files specified in the `path` parameter.

Parameter

`host`	IP address or host name of the FTP(S)/SFTP server **Possible values:** Any valid string (e.g. *`server.tld`*)
`port`	Port number of the FTP(S)/SFTP server **Possible values:** • Any positive integer • *`22`*: SFTP default port (default)
`user`	Username for the FTP(S)/SFTP connection
`password`	Password for the FTP(S)/SFTP connection
`path`	Path to a folder or file on the FTP(S)/SFTP server. **Possible values:** Any string (e.g. */Folder/Subfolder* oder */Folder/Subfolder/File.xml*)
`targetPath`	Additional path to a folder or file on the FTP(S)/SFTP server. ⚠ This parameter is used only in the `Move` operation. **Possible values:** Any string (e.g. `/Folder/Subfolder` oder `/Folder/Subfolder/File.xml`)
`protocol`	File transfer protocol **Possible values:** • *`FTP`*: Execute operations via *File Transfer Protocol* (default) • *`FTPS`*: Execute operations via *File Transfer Protocol Secure* (default) • *`SFTP`*: Execute operations via *SSH File Transfer Protocol* (default)
`connectMode`	Connection mode **Possible values:** • *`ACTIVE:`* Active connection mode • *`PASSIVE:`* Passive connection mode (default)
`nameFilter`	Filters the output of the operation `List` by name **Possible values:** • Any string (e.g. *`.xml`*) • Placeholder (wildcard) `*`

`addNewLine`	Inserts a new line in the destination file before the file is appended. Can only be used with the `Append File` operation. **Possible values:** • `None`: (default) Does not insert a new line. • `Auto`: Inserts a new line if no new line was inserted before. Detects the format of an already inserted new line. • `Windows`: Inserts a new line in Windows format. • `Unix`: Inserts a new line in Unix format.
`hostValidationEnabled`	Enable host validation to ensure that the requested SFTP server corresponds to the SFTP server that is being connected to. ⚠ The `hostValidationEnabled` parameter is only relevant for the SFTP protocol. **Possible values:** • *`Yes`*: Enable host validation (set flag) • *`No`*: Disable host validation (not recommended for production systems) (default)
`hostValidationFile`	Path to the *Known Hosts* file, which contains a list of confirmed hosts and their associated public keys. **Possible values:** Any valid path (e.g. *`C:\Data\known_hosts`*)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

✓ For more information, see the section

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

Operation: List

The `List` operation lists names and metadata of files and folders from the folder defined in the `path` parameter. Outputs an XML with metadata as a result.

`1`	The adapter operation was executed successfully.
`-1`	The folder doesn't exist.

Examples

The following example shows a sample parameterization of the `List` operation adapter to list PDF files that are in the folder defined in the `path` parameter.

Parameters	Value
`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>
```

When a folder is listed, the attribute `isDirectory="true"` is set.

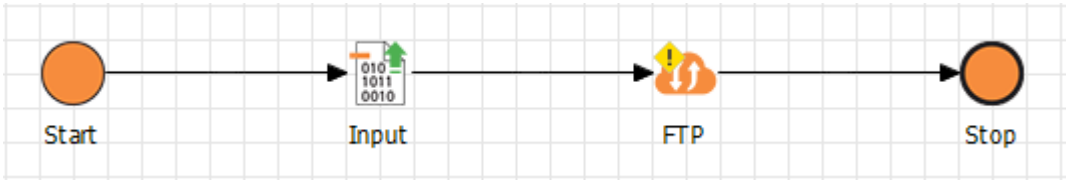
For further information please visit the section

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

Operation: Send File

The `Send File` operation writes data from the previous process step to the destination file defined in parameter `path`.

Examples



The following example shows a sample parameterization of the adapter for the `Send File` operation to write the data from the previous process step to the destination file defined in parameter `path`.

Parameters	Value
`path`	/MyFolder/MyFile.xml

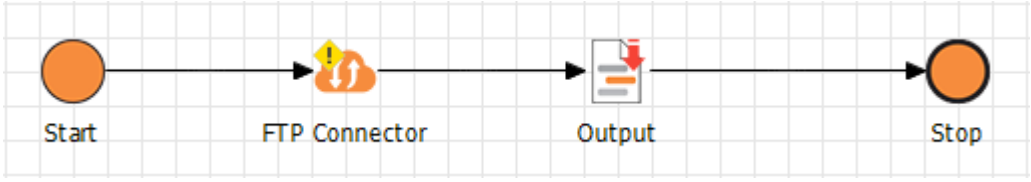
For further information please visit the section

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

Operation: Receive File

The `Receive File` operation reads the data from the file defined in the `path` parameter.

Examples



The following example shows a sample parameterization of the adapter for the `Receive File` operation to read the data from the destination file defined in the `path` parameter.

Parameters	Value
`path`	/MyFolder/MyFile.xml

The output of the adapter is then written to an XML file in the X4 repository.

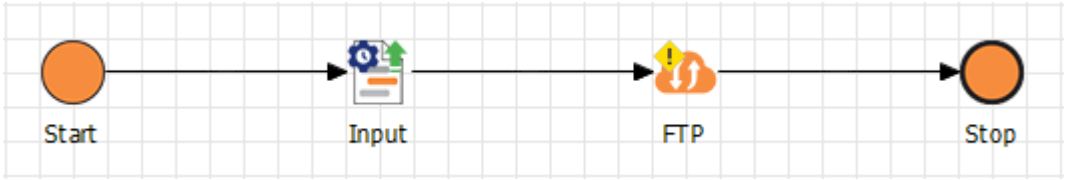
For further information please visit the section

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

Operation: Append File

The `Append File` operation appends the data from the previous process step to the destination file specified in parameter `path`.

Examples



The following example shows a sample parameterization of the adapter for the `Append File` operation to append the data from the previous process step to the destination file defined in `path` parameter.

Parameters	Value
`path`	/MyFolder/File.xml
`addNewline`	`Auto`

Target file

```
<?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>
```

The output is not a valid XML file. The output must be transformed into a valid XML file using a mapping.

For further information please visit the section

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

Operation: Move

The `Move` operation moves files or folders defined in the `path` parameter to the folder defined in the `targetPath` parameter.

Examples

Move folder
The following example shows a sample parameterization of the adapter for the `Move` operation to move the `MyFolder` folder to the `X4` folder:

Parameters	Value
`path`	/MyFolder
`targetPath`	/X4/MyFolder

Move file
The following example shows a sample parameterization of the adapter for the `Move` operation to move the `MyFile.xml` to the `X4` folder:

Parameters	Value
`path`	/MyFolder/MyFile.xml
`targetPath`	/X4/MyFile.xml

For further information please visit the section

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

Operation: Delete File

The `Delete File` operation deletes the file defined in the `path` parameter.

The data will be permanently deleted.

Examples

Delete file

The following example shows a sample parameterization of the adapter for the `Delete File` operation to delete the target file defined in the `path` parameter.

Parameters	Value
`path`	/MyFolder/MyFile.xml

For further information please visit the section

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

Operation: Create Folder

The `Create Folder` operation creates the folder defined in the `path` parameter.

Examples

The following example shows a sample parameterization of the adapter for the `Create Folder` operation to create the folder defined in the `path` parameter.

Parameters	Value
`path`	/MyFolder

For further information please visit the section

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

Operation: Delete Folder

The `Delete Folder` operation deletes the folder defined in the `path` parameter.

The data will be permanently deleted.

Examples

Delete folder
The following example shows a sample parameterization of the adapter for the `Delete Folder` operation to delete the destination folder specified in the `path` parameter.

Parameters	Value
`path`	/MyFolder

For further information please visit the section

- Operation: List
- Operation: Send File
- Operation: Receive File
- Operation: Append File
- Operation: Move
- Operation: Delete File
- Operation: Create Folder
- Operation: Delete Folder
- Operation: Exists

Operation: Exists

The `Exists` operation checks whether the files or folders specified in the `path` parameter exist. Outputs an XML with metadata as a result.

Status values

`1`	The operation was executed successfully.
`0`	The file or folder does not exist.
`-1`	The operation failed due to a technical error.

Examples

The following example shows a sample parameterization of the adapter for the `Exists` operation to check whether the destination file specified in parameter `path` exists.

Parameter	Value
`path`	/file.xml

Output
If the file exists:

```
<?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>
```

When checking if a folder exists, the attribute `isDirectory="true"` is set.

10.4.4.2 FTP Connector (Advanced)

`FTP Connector`: Supports FTP (*File Transfer Protocol*), SFTP (*SSH File Transfer Protocol*) and FTPS (*File Transfer Protocol Secure* or *FTP-SSL*) protocols for sending, receiving, deleting and moving one or more files and folders as well as listing folder contents. The adapter supports proxy servers of the type `HTTP`, `SOCKS4` and `SOCKS5`.

Former designation up to version 4.5.1: *File Transfer Adapter*

Properties

Operation (action)	Determines which file transfer operation is performed on the processed files or folders. **Possible values:** • `List`: Output the list of files in the folder specified in the `remotePath` parameter as an XML document. • `Send`: Save data from the previous process step to the folder specified in parameter `remotePath`, using the file name specified in parameter `documentName`. Or send multiple files from source folder in parameter `localPath` to `remotePath` by setting parameters `directoryMode` and `multipleFiles` to *true*; a filter expression can also be specified in parameter `filter`. • `Receive`: Read file into the process whose path has been defined in parameter `remotePath` and whose file name has been defined in parameter `documentName`. Or receive multiple files by setting the parameters `directoryMode` and `multipleFiles` to *true* and `localPath` to the local destination folder; a `filter` expression can also be specified in parameter filter. • `Delete`: Delete file or folder on the FTP(S)/SFTP server defined in parameter `documentName`, whose path is defined in parameter `remotePath`. To delete non-empty folders, additionally activate parameter `recurse`. • `Move`: Move a file or folder specified in the parameters from oldPath with `oldDocumentname` within the FTP(S)/SFTP server to `remotePath` and `documentname`. • `Make directory`: Create folder on the FTP(S)/SFTP server whose name is defined in parameter `documentName` and whose path is defined in parameter `remotePath`. • `Delete directory`: Delete folder on the FTP(S)/SFTP server whose name is defined in parameter `documentName` and whose path is defined in parameter `remotePath`.

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values**: *en.softproject.integration.adapter.transferadapter.TransferAdapter*: Main class (default)

`protocol`	File transfer protocol **Possible values:** • *`FTP`*: Execute operations via the *File Transfer Protocol (default)* • *`FTPS`*: Execute operations via *File Transfer Protocol Secure* • *`SFTP`*: Execute operations via the *SSH File Transfer Protocol*
`host`	IP address or host name of the FTP(S)/SFTP server **Possible values:** Any valid string (e.g. *`server.tld`*)
`port`	Port number of the FTP(S)/SFTP server **Possible values:** • Any positive integer • *`22`*: SFTP standard port (default)
`connectMode`	Connection mode **Possible values:** • *`ACTIVE`*: Active connection mode • *`PASSIVE`*: Passive connection mode (standard)
`controlEncoding`	Coding used for the command channel. This influences how umlauts are transmitted in file names. This value depends on the FTP server. **Possible values:** UTF-8 (Standard) If no `controlEncoding` is set or the `controlEncoding` corresponds to the empty string, the connection is not modified and the `controlEncoding` is undefined. Otherwise, the value is set as `controlEncoding` on the connection.
`login`	Username for the FTP(S)/SFTP connection
`password`	Password for the FTP(S)/SFTP connection; If parameter `privateKeyFile` is set: Password of the *private key*.
`accountInfo`	For additional information on the account, see also the FTP library documentation.

`loginCommandsUrl`	URL to the XML file in the XStore that contains the login commands. **Possible values:** Any XStore URL
`remotePath`	Path on the FTP(S)/SFTP server whose destination folder is to be handled via FTP(S)/SFTP. **Possible values:** Any string (e.g. *`folder/subfolder`*)
`documentname`	Name of the folder or file that will be processed. If `directoryMode=true` is set, the folder name is expected in the parameter. **Possible values:** Any string (e.g. *`Data.xml`*)
`oldPath`	Path to the folder or file on the FTP(S)/SFTP server to be moved (only for Operation `Move`). **Possible values:** Any character string (e.g. *`folder/subfolder`*)
`oldDocumentname`	Name of the file to be moved (only for Operation `Move`) **Possible values:** Any string (e.g. *`Data.xml`*)
`recurse`	Delete path recursively, i.e. including all subfolders and files (only with Operation `Delete`). **Possible values:** • `True`: Delete all included folders (and subfolders) and files. • `False`: Delete only folders and files in the current path (default)
`directoryMode`	Enable folder mode to process the content of folders; if parameter `multipleFiles` is enabled, all items matching the filter criterion in parameter filter are processed **Possible values:** • `true`: Process folder • `false`: Process single files only (default)
`localPath`	Local folder for sending or receiving multiple files (only if the parameters `directoryMode` and `multipleFiles` are activated) **Possible values:** Any valid path (e.g. *`C:\Directory`*)

`multipleFiles`	Process all files that match the filter criterion in parameter `filter` **Possible values:** • `true`: Process all applicable files • `false`: Process only the file specified in parameter `documentName`, or if parameter `directoryMode` is activated, the first file (if the filter criterion applies to multiple files) (default).
`filter`	Filter expression that can contain the wildcards *and ?to limit the amount of files processed. This applies to the operation `list` and the operations `receive` and `send` and only if the parameters `multipleFiles` and `directoryMode` are activated. **Possible values:** Any character string (e.g. *.xml*)
`deleteAfterReceive`	Delete file or folder after receiving from FTP(S)/SFTP server (only with Operation `Receive`) **Possible values:** • `true`: Delete the original file or folder after receiving it on the FTP(S)/SFTP server. • `false`: Keep original file or folder (default)
`deleteAfterSend`	Delete file or folder after sending to FTP(S)/SFTP server (only with Operation `Send`) **Possible values:** • `true`: Delete original file or folder after sending to FTP(S)/SFTP server. • `false`: Keep original file or folder (default)
`append`	Append sent data to the destination file if a destination file with the same name already exists (only with Operation `Send`). **Possible values:** • `true`: Append data in the target file (if available) • `false`: Overwrite destination file (if available) (default)
`transferMode`	Transmission type **Possible values:** • `BINARY`: Transfer as binary file (default) • `TEXT`: Transfer as text

`maxTransferRate`	Maximum transmission rate in bytes/second **Possible values:** • Any positive integer • `0`: no limit (default)
`sleepEnabled`	Activate *Sleep Mode* (if parameter `multipleFiles` is activated) **Possible values:** • `true`: Activate sleep mode (enables the definition of a rest period in the `sleepTime` parameter). • `false`: Disable sleep mode (default)
`sleepTime`	Waiting time in seconds between the transfer of the files (if parameters `multipleFiles` and `sleepEnabled` are activated). **Possible values:** • Any positive integer (default) • `0`: No use of rest time
`timeout`	Time limit for TCP connection establishment and transmission in milliseconds **Possible values:** • Any positive integer (e.g. *`50000`*) • `0`: No time limit (default)
`hostValidationEnabled`	Activate host validation to ensure that the requested FTP(S)/SFTP server also corresponds to the FTP(S)/SFTP server to which you are connecting. Ideally, deactivate for test purposes only! **Possible values:** • `true`: Enable host validation (set flag) If host validation is enabled, the `hostsFileName` parameter must be set. • `false`: Disable host validation (not recommended for productive systems) (default)

`implicitFTPS`	Enable Implicit FTPS **Possible values:** • `true`: Enable Implicit FTPS (set flag) • `false`: Disable Implicit FTPS (default)
`securityMechanism`	Safety mechanism **Possible values:** • `Explicit_TLS` • `Explicit_SSL` • `Explicit_TLS_C` • `Implicit_PRIVATE` • `Implicit_CLEAR`
`privateKeyFile`	Path to a *private key* file (not: *Java Keystore*). If you want to authenticate yourself on the FTP(S)/SFTP server with a *private key*, then the password parameter serves as the password for your *private key* file. **Possible values:** Any valid path (e.g. *`C:\Data\X4Keys\id_rsa`*)

`hostsFileName`	Path to the *Known Hosts* file, which contains a list of confirmed hosts and their associated public keys.
	
	Possible values: Any valid path (e.g. *`C:\Data\known_hosts`*)
	
	The *Known Hosts* file must be structured as follows:
	
	`host,IP Verschlüsselung Host_PublicKey`
	`ftp.softproject.de,217.160.223.49 ssh-rsa`
	`AAAAB3NzaC1yc2EAAAA...`

`useProxy`	Use proxy server to establish connection **Possible values:** • `true`: Use proxy server • `false`: Do not use proxy server (default)
`proxyType`	Proxy server type **Possible values:** • `HTTP`: Use HTTP proxy (default) • `SOCKS4`: Use proxy based on *SOCKS 4* protocol • `SOCKS5`: Use proxy based on *SOCKS 5* protocol

`proxyAddress`	IP address or host name of the proxy server **Possible values:** Any valid string (e.g. *`server.tld`*)
`proxyPublicAddress`	Public IP address or host name of the proxy server if the proxy does not return an address (required e.g. for active FTP transfers if the FTP(S)/SFTP server has to reconnect to the proxy). **Possible values:** Any valid string (e.g. *`server.tld`*)
`proxyPort`	Port number of the proxy server **Possible values:** Any positive integer
`proxyUsername`	Username for the proxy server
`proxyPassword`	Password for the proxy server

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

In file mode (if parameter `directoryMode` is deactivated), the `Send` operation expects an input document to be sent.

The different login operations have an order. If the respective parameters are filled, applies:

1. Login commands are used
2. User/Password with `AccountInfo` is used
3. User/password is used

Definition of the LoginCommands XML:

```
<loginCommands> <loginCommand>cmd1</loginCommand> <loginCommand>cmd2</loginCommand>
</loginCommands>
```

Output

In file mode (if parameter `directoryMode` is deactivated), the received document is output for the operation `Receive`.

For Operation `List`, an XML document with the list of folders and files is output.

10.4.5 iCalendar

? Unknown Attachment `iCalendar`: Creates an *iCalendar* event from an adapter-specific XML document according to RFC 5545 or converts an *iCalendar* text document into an XML document. In addition, a *POP/SMTP* adapter can be dynamically parameterized so that an *iCalendar* event (e.g., an appointment invitation) is sent as an attachment to an e-mail.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *`iCal To XML`*: Convert *iCal* text document to XML • *`XML To iCal`*: Convert adapter-specific XML structure to an *iCalendar* text document. • *`Create iXmap for E-Mail Adapter`*: Output *iXmap* XML structure with parameter values and the *iCal* event as e-mail body, which is used to dynamically parameterize a `POP/SMTP adapter`; see *X4 Developer Reference - Dynamic Processes via iXmap.* ⚠ This operation will be deprecated with the next release.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.icalendar.ICalendarAdapter*: Main class (default)
attachmentName	File name of the email attachment with the *iCal* calendar item (for `Create iXmap for E-Mail Adapter operation`). **Possible values:** Any valid file name

Status values

1	The operation was executed successfully.

`-1`	The operation failed due to a technical error.

Example

In the demo system *X4 Experience* you can find the sample process `iCalendarAdapter.wrf` for using the adapter.

Dependencies

This adapter the following libraries:

- `ical4j-1.0.jar`
- `backport-util-concurrent-3.1.jar`
- `icalendar.jar`

10.4.5.1 Operation: iCal to XML

Operation `iCal To XML` can be used to convert a valid *iCal* text document to XML.

Input

For Operation `iCal To XML`, the adapter expects any valid *iCal* text document.

Output

For Operation `iCal To XML`, the adapter outputs an XML document containing all appointment information (see the description of the input structure for Operation `XML To iCal`).

10.4.5.2 Operation: XML to iCal

Operation `XML To iCal` can be used to convert an adapter-specific XML structure into an *iCalendar* text document.

Input

The adapter expects an input XML document according to the following pattern:

```
<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>
```

Output

For Operation `XML To iCal`, the adapter outputs a corresponding *iCalendar* text document.

10.4.5.3 Operation: Create iXmap for E-Mail Adapter

> ⚠ *This operation will be deprecated with the next release.*

Operation `Create iXmap for E-Mail Adapter` can be used to output an *iXmap XML* structure with parameter values and the *iCal* event as e-mail body, which is used to dynamically parameterize a *POP/SMTP* adapter; see *X4 Developer Guide - Dynamic Processes via iXmap*.

Input

For Operation `Create iXmap for E-Mail Adapter` expects the same XML structure as for Operation `XML To iCal`.

In addition, however, the root element can contain an `<mail>` element as the first child element, in which an element of the same name, including value, can be defined for each parameter that is to be set dynamically at the *POP/SMTP* adapter:

```
<iCalendarData>
   <mail>
      <!-- für jeden zu setzenden Parameter -->
      <Parametername 1><!--Wert 1--></Parametername 1>
      <Parametername n><!--Wert n--></Parametername n>
   </mail>
   <!-- Termindaten -->
</iCalendarData>
```

Output

For Operation `Create iXmap for E-Mail Adapter`, the adapter outputs an *iXmap* XML document that is used to dynamically parameterize a *POP/SMTP* adapter; see *X4 Developer Guide - Dynamic Processes via iXmap*.

However, in order for the process engine to recognize the *iXmap* structure as such, this XML document must be transformed again via XSL (identity transformation is sufficient). A subsequent *POP/SMTP* adapter is then dynamically parameterized with the values specified in the input XML document.

```
<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`: Pings an IP address via the *Internet Control Message Protocol*. In doing so, the adapter sends *ICMP ECHO REQUEST*s or alternatively tries to establish a TCP connection on port 7. If previous connection requests fail and the port is set to a value other than 7, another TCP request is sent to the specified port. The adapter outputs an XML document containing the result of the echo (*`true`* or *`false`*).

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *`Ping`*: Execute ping

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.icmpping.ICMPPing`*: Main class (default)
iPAddress	IP address to which the ping is sent **Possible values:** • Any IP address (e.g. *`127.0.0.1`*) • *`0.0.0.0`*: Empty IP address (default)
timeout	Time limit for establishing connection in milliseconds **Possible values:** • Any positive integer • *`0`*: No time limit (default)
port	Port to retry connection if the first ICMP request fails **Possible values:** • Any positive integer

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

10.4.7 SCP

`SCP`: Sends and receives individual files via SCP (*Secure CoPy*). The adapter supports HTTP, SOCKS4 and SOCKS5 type proxy servers.

Former designation until version 4.5.1: *SCP Transfer Adapter*

Properties

Operation (action)	Determines which file transfer operation is performed on the processed files or folders. **Possible values:** • *`Send`*: Send a file specified in parameter `documentname`, located in a local folder specified in parameter `localPath`, to a remote folder specified in parameter `remotePath`. • *`Receive`*: receive file whose path was defined in `remotePath` parameter and whose filename was defined in `documentName` parameter and save it in a local folder specified in `localPath` parameter.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.scpadapter.SCPAdapter`* : Main class (default)
host	IP address or host name of the remote computer **Possible values:** Any valid string (e.g. *`server.tld`*)
port	Port number of the remote computer **Possible values:** • Any positive integer • *22*: SCP default port (default)
login	Username for the SCP connection

password	Password for the SCP connection; If parameter `privateKeyFile` is set: Password of the *Private Key*.
remotePath	Path on the remote computer with which a file is to be exchanged **Possible values:** Any string (e.g. *`folder/subfolder`*)
documentname	Name of the file to be sent or received **Possible values:** Any string (e.g. *`Data.xml`*)
localPath	Local folder for sending and receiving a file **Possible values:** Any valid path (e.g. *`C:\Directory`*)
timeout	Time limit for TCP connection establishment and transmission in milliseconds **Possible values:** • Any positive integer (e.g. *`50000`*) • *`0`*: No time limit (default)
hostValidationEnabled	Enable host validation to ensure that the requested remote host also corresponds to the host to which you are connecting. Ideally, deactivate for test purposes only! **Possible values:** • *`Yes`*: Enable host validation (set flag) • *`No`*: Disable host validation (not recommended for production systems)(default)
privateKeyFile	Path to a *Private Key* file (not: *Java Keystore*). If you want to authenticate on the remote machine with a *Private Key*, then the `password` parameter serves as the password for your *Private Key* file. **Possible values:** Any valid path (e.g. *`C:\Data\X4Keys\id_rsa`*)
hostsFileName	Path to the *Known Hosts* file, which contains a list of confirmed hosts and their associated public keys. **Possible values:** Any valid path (e.g. *`C:\Data\known_hosts`*)

useProxy	Use proxy server to establish connection **Possible values:** • *Yes*: Use proxy server • *No*: Do not use a proxy server (default)
proxyType	Proxy server type **Possible values:** • *HTTP*: Use HTTP proxy (default) • *SOCKS4*: Use proxy based on *SOCKS 4* protocol • *SOCKS5*: Proxy auf Basis des *SOCKS 5*-Protokolls verwenden
proxyAddress	IP address or host name of the proxy server **Possible values:** Any valid string (e.g. *server.tld*)
proxyPublicAddress	Public IP address or host name of the proxy server if the proxy does not return an address. **Possible values:** Any valid string (e.g. *server.tld*)
proxyPort	Port number of the proxy server **Possible values:** Any positive integer
proxyUsername	Username for the proxy server
proxyPassword	Password for the proxy server

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

Input

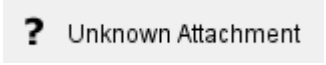
The adapter does not expect a specific input document. An input document is output unchanged.

Output

An input document is output again unchanged.

10.4.8 TCP/IP Connector/Listener

10.4.8.1 TCP/IP Communication



`TCP/IP Communication`: Executes synchronous or asynchronous TCP socket calls, opening a socket and sending the input data to the specified host.

Properties

Operation	Determines which operation the adapter executes **Possible values:** `Approach`: Call TCP socket (default)

Parameter

Adapter	Main class of the adapter (do not change!) **Posspible values:** *`de.softproject.integration.adapter.socketapproach.SocketApproach`*: Main class (default)
iPAddress	IP address of the host computer to which data is to be sent **Possible values:** • Any IP address (e.g. *`127.0.0.1`*) • *`0.0.0.0`*: Empty IP address (default)
port	Port of the host computer **Possible values:** • Any positive integer (e.g. *`8080`*) • *`0`*: Port 0 (default)
timeout	Time limit for establishing connection in milliseconds **Possible values:** • Any positive integer • *`0`*: No time limit (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.4.9 Telnet Connector

`Telnet Connector`: Connects to servers via the Teletype Network Protocol.

Properties

`Operation`	Operation performed by the adapter **Possible values**: `Execute`: Executes the commands available via an input XML

Parameter

`Adapter`	Adapter main class (do not change!) **Possible values:** *`en.softproject.integration.adapter.telnet.TelnetAdapter`*: Main class (default)
`host`	*Required parameter*: Host of the server to which the connection is to be established **Possible values:** • `IP`: Any valid host IP, e.g. *`1.1.1.1`* • `String`: Any valid host name, e.g. *`myServer.com`*
`port`	Port of the server to which the connection is to be established **Possible values:** • Any integer • `23` (default): Default port for Telnet connections
`user`	User name for the Telnet connection **Possible values**: String with the user name, e.g. *`MyUser`*
`password`	Password for the Telnet connection **Possible values**: String with the Password, e.g. *`MyPassword`*
`timeout`	Connection timeout in milliseconds **Possible values:** • Any integer • `30000` (default): Default value of 30 seconds

`responseDelay`	Response delay in milliseconds **Possible values:** • Any integer • `750` (default): Default value of 0.75 seconds
`cleanAnsiVTSequences`	Specify whether the output should contain ANSI VT characters or not **Possible values:** • `true` (default): The output contains ANSI VT characters • `false`: The output does not contain ANSI VT characters

Status values

`1`	The adapter operation was successfully executed
`-1`	An error occurred during the execution of the adapterThe adapter operation was executed successfully

Input

When connecting via Telnet, the adapter uses labels within the `login`: and `password`: credentials by default. However, if you want to connect to a machine that has different access labels, these can be configured in the input XML via the `<Configuration>` node.

The `Telnet Connector` expects the following input structure:

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>
```

Explanation:

- `Telnet:` Root node
- `Configuration:` Node with the same content as specified in the input. Although the node is required, the `loginLabel` and `passwordLabel` attributes are optional.
- `Commands:` node that includes all commands used
- `Command:` Any number of `command` nodes with a `value` attribute containing the commands to be executed.

Output

The `Telnet Connector` outputs an XML structure similar to the following output:

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>
```

Explanation:

- `Telnet`: Root node
- `LoginResult`: Information about the login result. The value of the node is either `Failed` or `Successful`.
- `Configuration`: Node with the same content as specified in the input.
- `Commands`: node that includes all commands used
- `Command`: Any number of `command` nodes with a `value` attribute containing the commands to be executed. The content of this node is the result of the command issued by the Telnet server.

10.5 Compression

- gzip Converter
- ZIP

10.5.1 gzip Converter

? Unknown Attachment `gzip Converter`: Compresses and decompresses input data using the *gzip* algorithm.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *Encode*: Compress data gzip • *Decode*: Decompress gzip compressed data as *MIME type application/octet stream*

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.GZipAdapter*: Main class (default)

Status values

1	The operation was executed successfully.
-1	The operation failed due to a technical error.

See also

- ZIP in Memory Compression (Deprecated)
- Adapter

10.5.2 ZIP

10.5.2.1 Zip Converter

`Zip Converter`: Enabels to zip and unzip files and folders from input (in memory) or file system.

Properties

`Operation`	Defines the operation executed by the adapter. **Possible values:** • `Zip`: Compresses the provided input in a zip file. Encryption and compression settings can be set. **Required parameters** Parameters `SourcePath` or `InputName` are required for this operation. • `Unzip`: Decompresses the provided input to the specified location. Password protected files can also be decompressed.

Parameter

`Adapter`	Main class of the adapter (Do not change!) **Possible values:** *de.softproject.x4.adapter.zipconverter.ZipConverter*: Main class (default)
`sourcePath`	File system path to the file to be compressed/decompressed. **Possible values:** • File system path If this parameter is not set, the file is provided by the adapter input. In this case parameter `inputName` is required for operation `Zip`.
`targetPath`	File system path to the memory location where the output is to be stored ("target file system"). **Possible values:** • File system path If this parameter is not set, the file is provided by the adapter output.

`overwrite`	Defines if existing files with the same name should be overwritten. **Possible values:** • Checked ⓘ • If `mergeContent` is enabled, the output is an error, because the file can not be processed. • If `mergeContent` is disabled, the existing file is overwritten. • Unchecked ⓘ • If `mergeContent` is enabled, the files are merged. • If `mergeContent` is disabled, the output is an error, because the file can not be processed.
`mergeContent`	Defines if existing files with the same name should be merged. **Possible values:** • Checked ⓘ • If `overwrite` is enabled, the output is an error, because the file can not be processed.. • If `overwrite` is disabled, the files are merged. • Unchecked ⓘ • If `overwrite` is enabled, the existing file is overwritten. • If `overwrite` is disabled, the output is an error, because the file can not be processed.
`inputName`	Name of the input file. Is used if parameter `sourcePath` is not set. **Possible values:** • File name of the file, that should be received as input or provided as output.
`encryptedFiles`	Defines if encrypted files should be processed in the operation. With operation `Zip`, the zip file output is password protected, with operation `Unzip`, a password is required for unzipping. **Possible values:** • Checked / Unchecked
`password`	Password for encrypted files. **Possible values:** Any string

`encryptionMethod`	Defines which encryption method is used for zipping when `encryptedFiles` is checked. **Possible values:** • `No encryption`: Do not use encryption when zipping. • `AES`: Use AES encryption algorithm (Key size: 256 Bit) • `Standard`: Use standard encryption algorithm
`compressionMethod`	Definies which compression algorithm is used for zipping. **Possible values:** • `Deflate` (Default): Lossless compression using a combination of the LZ77 algorithm and Huffmann encoding. Please find more information on this compression level here and here. ⓘ If `Deflate` is selected, the compression level can be specified with `CompressionLevel`. • `Store`: Data is not compressed, but packed into a single file. ⓘ This option is useful when multiple data/folders are to be sent/received and the file size can be neglected. No time is needed for packaging.
`compressionLevel`	Defines which compression algorithm when zipping if `Deflate` is selected for `compressionMethod`. **Possible values:** • `Fastest`: Lowest compression level, but very fast compression • `Fast`: Low compression level, but fast compression • `Normal` (Default): Optimum ratio between compression level and speed • `Maximum`: High compression level, but slow compression • `Ultra`: Hightest compression level, but very slow compression

Status values

`1`	The operation was executed successfully.
`-1`	An error occurred during the operation's execution.

Input

The adapter expects input with following structure:

- Both operations (`Zip` / `Unzip`) are able to work with two different places of origin or target.
  - Input/Output
  - File system path for Input/Output
- Based on these two possibilities, different inputs can be used for both operations:
  - Parameter `sourcePath` is set: Input from the directory specified in the file system path.

    - Parameter `sourcePath` is not set: Input of the parameter
- Based on these two possibilities, different outputs can be used for both operations:
    - Parameter `targetPath` is set: The result of the operation is stored in the location specified as file system path.
    - Parameter `targetPath` is not set:
        - An XML file with one entry for each input file contained in the input zip file is output.
        - The contents of the zip file are stored in the location specified as the directory path.

Output

Depending on the executed operation, the adapter returns different kinds of documents.

- Operation `Zip` without `targetPath`: Zip file is provided in adapter output.
- Operation `Zip` with `targetPath`: Zip file is stored in the target directory.
- Operation `Unzip` without `targetPath`: An XML file that contains the various files within the zip file is provided in the adapter output. The XML file has the following structure:

Sample output for operation Unzip

```
<?xml version="1.0" encoding="UTF-8" ?>
<Zip>
    <File size="209" name="testToZip.xml" comment="&#0;&#0;!
>PD94bWwgdmVyc2lvgj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPFJvb3RFbGVtZW50P</File>
</Zip>
```

- Operation `Unzip` with `targetPath`: The content of the zip file is stored in the target directory.

10.6 Databases

- JDBC Connector
- JDBC Iterator
- JDBC Metadata Explorer

10.6.1 JDBC Connector

10.6.1.1 About the JDBC Connector

`JDBC Connector`: Connects to a JDBC-compatible database and executes one or multiple database statements handed over as input XML document with an adapter-specific structure.

Former name until version 4.5.1: *JDBC Database Connection Adapter*

The connector requires a JDBC datasource that has been deployed within the application server and for which the connection URL, credentials, schema and transaction behavior etc. are defined. For further information on deploying JDBC datasources consult your application server's documentation.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** `execute` : Execute the function adapter

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.jdbc.DB*: Main class (Default)
JndiName	JNDI name of the DataSource If the connection is established via the JNDI name, the parameter `connectionURL` must not be specified. **Possible values:** Valid JNDI name (e. g. `java:/DefaultDS`)

ColumnCase	Controls how column names of the result set are copied into the result XML document. Since databases have different behaviour regarding the case of columns in the result set, the processing of the output of this adapter can become independent of that behaviour by using the `ColumnCase` parameter. **Possible values:** • *NO_CHANGE*: Does not change the column names, i. e. the column names will be written to the result document as returned by the database (Default) • *TO_LOWER*: Changes the column names to lower case • *TO_UPPER*: Changes the column names to upper case ⓘ Column names, which are illegal xml element names (`NCNAME`) are converted to `_<number>` (e. g. `_2`) where *<number>* is the column index.
connectionURL	Connection URL for the JDBC database ⓘ If the connection is established via the connection URL, the parameter `JndiName` must not be specified. **Possible values:** • The URLs structure depends on the database to which a connection is to be established. • Example for sqlite: *jdbc:sqlite:C:/Temp/mydatabase.db*
user	User name for authentication via the parameter `connectionURL`
password	Password for authentication via the parameter `connectionURL`
rootElementName	Name of the root element of the output **Possible values:** String that is valid for XML element names. Default: *Data*

CommitMode	Defines the version to be used for commits **Possible values:** • *NONE*: Do not execute any commit • *AUTO*: Automatic commit according to the JDBC API (i. e. after each batch) (Default) • *STATEMENT*: Explicit commit after each statement (i. e. after all batches for this statement) • *DOCUMENT*: At the end of the document (i. e. after executing all statements) • *MANAGED*: Use adapter in transactions (for externally managed transactions)
AddMetaDataForQueries	(with X4 4.1 Revision 32408 or higher) Controls whether database metadata should be output for each *result set* in the result XML document of the adapter, when an SQL query is executed (not available for `UPDATE, DELETE` etc.). In the JDBC adapter's batch mode (`StatementBatch, PreparedStatementBatch`, and `CallableStatementBatch`) this function is also not available. **Possible values:** • *true*: Add database metadata to result document • *false*: Output result document without database metadata (Default)

Status values

`1`	The query has returned data records or the instruction has changed data records. The status `1` is returned, if the document contains multiple queries or instructions and at least one query has delivered or changed a result.
`0`	The query did not return any results or the statement hasn't changed any data records. The status `0` is returned, if the document contains multiple queries or instructions and no query has returned or changed a result.

-1	Technical error: more information on the error cause can be found in the server log.

Input

The *JDBC Connector* excepts specific input XML structures containing database statements. These input XML structures can be generated dynamically by using XSL mappings that depend on the processed XML data.

See also:

- Statement: Direct SQL
- StatementBatch: Direct SQL in Batch Mode
- PreparedStatement: SQL with Values
- PreparedStatementBatch: SQL with Values in Batch Mode
- CallableStatement: Stored Procedures
- CallableStatementBatch: Stored Procedures in Batch Mode
- NamedPreparedStatement: SQL with Named Parameters
- NamedPreparedStatementBatch: SQL with Named Parameters in Batch Mode
- Passing XML Structures
- Retrieving Database Metadata

10.6.1.2 Statement: Direct SQL

Input

```
<any element name>
   <statement resulttag="result element name" timezone="timezone" format="Data
format">
      <sql> SQL statement </sql>
   </statement>
</any element name>
```

The following rules apply to each `<statement/>` element:

- The optional attribute `timezone` specifies a timezone according to `java.util.TimeZone`, e. g. *`GMT+2`*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`.
- It contains at least one `<sql/>` element with an SQL statement.

If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]>`

Example

```
<select>
   <statement resulttag="result" timezone="UTC" format="dd.MM.yyyy">
      <sql> <![CDATA[SELECT * FROM JMS_USERS]]> </sql>
   </statement>
</select>
```

Output

If the SQL statement was executed successfully, the database connection element will output an XML document containing the query response as a string with the following structure:

```
<Data>
   <result-element-name>
      <Field1><!--value--></Field1>
      <Fieldn><!--value--></Fieldn>
   </result-element-name>
</Data>
```

For `SELECT` statements used to access database tables with column names that do not represent a valid XML name, synthetic element names are created within the result XML document, in order to ensure that the XML document is well-formed.

10.6.1.3 StatementBatch: Direct SQL in Batch Mode

Input

In batch mode, the *JDBC Connector* executes multiple SQL statements at the same time, if they are deposited in the following input XML structure. In this case, only SQL statements are possible, which do not return a ResultSet, i.e. no queries with `SELECT`:

```
<any element name>
   <statementbatch resulttag="result element name" timezone="timezone"
    format="Date format">
      <sql> SQL statement 1</sql>
      <sql> SQL statement n</sql>
   </statementbatch>
</any element name>
```

The following rules apply to each `<statementbatch/>` element:

- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`.
- It contains at least one `<sql/>` element that each contains an SQL statement.

If you want to use characters within the SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=100 ]]>`

Example

```
<select>
   <statementbatch resulttag="result-element-name" 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

If the batch statement was executed successfully, the database connection element will output the following structure:

```
<Data>
   <result-element-name>
      <!--status of the batch operation-->
   </result-element-name>
</Data>
```

The following status outputs are possible for the Batch operation:

- *(number of rows)* `rows affected.` or
- `SUCCESS_NO_INFO` oder
- `EXECUTE_FAILED`

10.6.1.4 PreparedStatement: SQL with Values

The *JDBC Connector* executes SQL statements with dynamical input data, if they are deposited in the following input XML structure.

With X4 BPMS release 5.0 or higher, you can alternatively execute SQL statements using named parameters, see NamedPreparedStatement: SQL with Named Parameters.

Input

```
<any element name>
   <preparedstatement resulttag="result element name" timezone="time zone"
    format="date format">
      <sql>
         <!--SQL statement-->
      </sql>
      <columns>
         <column mode="in" name="any NC name"
          index="placeholder index" type="data type"
          timezone="time zone" format="date format">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--value--></Name>
         </row>
      </rowset>
   </preparedstatement>
</any element name>
```

The following rules apply to each `<preparedstatement/>` element:

- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`.
- It contains at least one `<sql/>` element.

> If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]>`

The following rules apply to each `<column/>` element:

- It can be used as often as required (`0-n` times)
- The optional attribute `mode` indicates the "direction" of the data stream (*in*: Input parameter for the database; *out*: Result parameter from the database; *inout*: Both, input parameter and result parameter; default: *in*.
- The attribute `name` specifies any NC name for a parameter.
- The attribute `index` specifies an integer placeholder index for the parameter defined within the attribute `name`.
- The optional attribute `type` specifies the data type (e. g. *INTEGER* or *BLOB; Default: VARCHAR*).
- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`.

The following rules apply to each `<rowset/>` element:

- It can be used as often as required (`0-n` times)
- It contains at least one element `<row/>` controlling the statements' execution.

The following rules apply to each `<row/>` element:

- It can contain any number of elements with the column name or the column alias set in the SQL statement.
- The elements inside contain the respective value.

Please note

- The attributes `mode` and `format` are required parameters up till X4 Server Revision 31469; in newer version they are optional.
- Allowed data types for the attribute `type` when setting (input parameter) and reading (*ResultSet*) values are: *`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`* and *`VARCHAR`* (default), *`XML`* (alias for SQLXML).

Example

```
<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

If the SQL statement was executed successfully, the adapter outputs an XML document containing the query response as a string with the following structure:

```
<Data>
   <result element name>
      <Field1><!--value--></Field1>
      <Fieldn><!--value--></Fieldn>
   </result element name>
</Data>
```

10.6.1.5 PreparedStatementBatch: SQL with Values in Batch Mode

In batch mode, the *JDBC Connector* executes multiple SQL statements at the same time, if they are deposited in the following input XML structure. In this case, only SQL statements are possible, which do not return a ResultSet, i.e. no queries with `SELECT`.

> ⓘ With X4 BPMS release 5.0 or higher, you can alternatively execute SQL statements in batch mode using named parameters, see NamedPreparedStatementBatch: SQL with Named Parameters in Batch Mode.

Input

```
<any element name>
   <preparedstatementbatch resulttag="result element name" timezone="timezone"
 format="date format">
      <sql>
         <!--SQL statement-->
      </sql>
      <columns>
         <column mode="in" name="any NC name"
          index="placeholder index" type="data type"
          timezone="time zone" format="date format">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--value--></Name>
         </row>
         <!-- Multiple <row> elements for batch mode -->
      </rowset>
   </preparedstatementbatch>
</any element name>
```

The following rules apply to each `<preparedstatementbatch/>` element:

- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`.
- It contains at least one `<sql/>` element.

If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=? ]]>`

The following rules apply to each `<column/>` element:

- It can be used as often as required (`0–n` times)
- The optional attribute `mode` indicates the "direction" of the data stream (*`in`*: Input parameter for the database; *`out`*: Result parameter from the database; *`inout`*: Both, input parameter and result parameter; default: *`in`*.
- The attribute `name` specifies any NC name for a parameter.
- The attribute `index` specifies an integer placeholder index for the parameter defined within the attribute `name`.
- The optional attribute `type` specifies the data type (e. g. *`INTEGER`* or *`BLOB; Default: VARCHAR`*).
- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *`GMT+2`*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`.

The following rules apply to each `<rowset/>` element:

- It can be used as often as required (`0–n` times)
- It contains any number of `<row/>` elements controlling the statements' execution.

The following rules apply to each `<row/>` element:

- It can contain any number of elements with the column name or the column alias set in the SQL statement.
- The elements inside contain the respective value.

Please note

- The attributes `mode` and `format` are required parameters up till X4 Server Revision 31469; in newer version they are optional.
- Allowed data types for the attribute `type` when setting (input parameter) and reading (*ResultSet*) values are: *`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`* and *`VARCHAR`* (default), *`XML`* (alias for SQLXML).

Example

```
<Rootelement>
   <preparedstatementbatch resulttag="result-element-name" 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

If the batch statement was executed successfully, the database connection element outputs the following XML structure:

```
<Data>
   <result-element-name>
      <!--status of the batch operation-->
   </result-element-name>
</Data>
```

The following status outputs are possible for the batch operation:

- *(number of rows)* `rows affected`. or
- `SUCCESS_NO_INFO` or
- `EXECUTE_FAILED`

10.6.1.6 CallableStatement: Stored Procedures

The *JDBC Connector* executes stored procedures via *JDBC callable statements* with dynamical input data, if they are deposited in the following input XML structure.

Input

```
<any element name>
   <callablestatement resulttag="result element name" timezone="timezone"
    format="date format">
      <sql>
        <!--SQL statement -->
      </sql>
      <columns>
         <column mode="mode" name="any name" index="placeholder"
          type="data type" timezone="time zone" format="date format">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--value--></Name>
         </row>
      </rowset>
   </callablestatement>
</any element name>
```

The following rules apply to each `<callablestatement/>` element:

- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*; see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a data format according to `java.text.SimpleDateFormat`.
- It contains at least one `<sql/>` element.

> If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]>`

The following rules apply to each `<column/>` element:

- It can be used as often as required (`0–n` times)
- The optional attribute `mode` indicates the "direction" of the data stream (*in*: Input parameter for the database; *out*: Result parameter from the database; `inout`: Both input parameter and result parameter; see http://download.oracle.com/javase/1,5.0/docs/guide/jdbc/getstart/callablestatement.html); Default: *in*.
- The attribute `name` specifies an arbitrary name for a parameter.
- The attribute `index` specifies an integer placeholder index for the parameter defined within the attribute `name`.
- The optional attribute `type` specifies the data type (allowed data types see note below; default: *VARCHAR*).
- The attribute `name` specifies an arbitrary name for a parameter.
- The optionale Attribut `timezone` specifies a time zone according to `java.util.TimeZone`, e.g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.

- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat` an.

The following rules apply to each `<rowset/>` element:

- It can be used as often as required (*0–n* times).
- It contains any number of `<row/>` elements controlling the statements' execution.

The following rules apply to each `<row/>` element:

- It can contain any number of elements with the column name or the column alias set in the SQL statement.
- The elements inside contain the respective value.

Please note

- The attributes `mode` and `format` are required parameters up till X4 Server Revision 31469; in newer version they are optional.
- Allowed data types for the attribute `type` when setting (input parameter) and reading (*ResultSet*) values are: *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* and *VARCHAR*(default), *XML* (alias for SQLXML).

Example

```
<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

If the SQL statement was executed successfully, the database connection element outputs an XML document containing the query response as a string with the following structure:

```
<Data>
   <<result element name>
      <Feld1><!--value--></Feld1>
      <Feldn><!--value--></Feldn>
   </<result element name>
</Data>
```

10.6.1.7 CallableStatementBatch: Stored Procedures in Batch Mode

In batch mode, the *JDBC Connector* executes stored procedures via *JDBC callable statements* with dynamical input data, if they are deposited in the following input XML structure. Only SQL statements are feasible that do not return any ResultSet, for example no queries with a `SELECT`.

Input

```
<any element name>
   <callablestatementbatch resulttag="result element name" timezone="timezone"
 format="date format">
      <sql>
        <!--SQL statement -->
      </sql>
      <columns>
         <column mode="mode" name="any name" index="placeholder" type="data type"
 timezone="timezone" format="date format">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--value--></Name>
         </row>
      </rowset>
   </callablestatementbatch>
</any element name>
```

The following rules apply to each `<callablestatementbatch/>` element:

- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*; see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a data format according to `java.text.SimpleDateFormat`.
- It contains at least one `<sql/>` element.

> If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]`

The following rules apply to each `<column/>` element:

- It can be used as often as required (`0–n` times)
- The optional attribute `mode` indicates the "direction" of the data stream (*`in`*: Input parameter for the database; *`out`*: Result parameter from the database; `inout`: Both input parameter and result parameter; see http://download.oracle.com/javase/1,5.0/docs/guide/jdbc/getstart/callablestatement.html); Default: *`in`*.
- The attribute `name` specifies an arbitrary name for a parameter.
- The attribute `index` specifies an integer placeholder index for the parameter defined within the attribute `name`.
- The optional attribute `type` specifies the data type (allowed data types see note below; default: *`VARCHAR`*).
- The attribute `name` specifies an arbitrary name for a parameter.
- The optionale Attribut `timezone` specifies a time zone according to `java.util.TimeZone`, e.g. *`GMT+2`*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat` an.

The following rules apply to each `<rowset/>` element:

- It can be used as often as required (*`0–n`* times).
- It contains any number of `<row/>` elements controlling the statements' execution.

The following rules apply to each `<row/>` element:

- It can contain any number of elements with the column name or the column alias set in the SQL statement.
- The elements inside contain the respective value.

Please note

- The attributes `mode` and `format` are required parameters up till X4 Server Revision 31469; in newer version they are optional.
- Allowed data types for the attribute `type` when setting (input parameter) and reading (*ResultSet*) values are: *`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`* and *`VARCHAR`* (default), *`XML`* (alias for SQLXML).

Example

```
<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

If the batch statement was executed successfully, the database connection element outputs the following XML structure:

```
<Data>
   <result element name>
      <!--status of the batch operation-->
   </result element name>
</Data>
```

The following status outputs of the batch operation are possible:

- *(number of rows)* `rows affected.` or
- `SUCCESS_NO_INFO` or
- `EXECUTE_FAILED`

10.6.1.8 NamedPreparedStatement: SQL with Named Parameters

The *JDBC Connector* executes SQL statements with named parameters, if they are deposited in the following input XML structure.

Input

```
<any element name>
   <namedpreparedstatement resulttag="result element name" timezone="time zone"
    format="date format">
      <sql>
         <!--SQL statement (optionally within a CDATA block) -->
         <!--Each parameter has an XML element with the same name as the
corresponding <row> child element -->
      </sql>
      <rowset>
         <row>
            <Name><!--value--></Name>
         </row>
      </rowset>
   </namedpreparedstatement>
</any element name>
```

The following rules apply to each `<namedpreparedstatement/>` element:

- The optional `timezone`attribute specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat` and defines the returned date values' formats.
- It contains one `<sql/>` element.

The following rules apply to each named parameter XML element within element `<sql/>`:

- The element name for the parameter is equivalent to the element name within the corresponding `<row/>` element.
- The optional attribute `mode` indicates the "direction" of the data stream (*in*: Input parameter for the database; *out*: Result parameter from the database; *inout*: Both, input parameter and result parameter; default: *in*.
- The optional attribute `type` specifies the data type, e. g. *INTEGER*, *BLOB,* or *VARCHAR*(default).
- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`. and defines the expected value format of the corresponding `<row>` element.

If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ SELECT * FROM TABLE WHERE ID<]]><id type="INTEGER" />`.

The following rules apply to each `<rowset/>` element:

- It can be used as often as required (`0-n` times)
- It contains any number of `<row/>` elements controlling the statements' execution.

The following rules apply to each `<row/>` element:

- It can contain at least one elements with a name that corresponds to the parameter name in the SQL statement.
- The elements inside contain the corresponding value.

For each data element within `<row/>` make sure to avoid unintended line breaks within the element value, as all line breaks would be converted to `\n` in the generated SQL statement.

Please note
Allowed data types for the attribute `type` when setting (input parameter) and reading (*ResultSet*) values are: *`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`* and *`VARCHAR`* (default), *`XML`* (alias for SQLXML).

Example

```
<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

If the SQL statement was executed successfully, the database connection element outputs an XML document containing the query response as a string with the following structure:

```
<Data>
   <result element name>
      <Field1><!--value--></Field1>
      <Fieldn><!--value--></Fieldn>
   </result element name>
</Data>
```

10.6.1.9 NamedPreparedStatementBatch: SQL with Named Parameters in Batch Mode

The *JDBC Connector* executes multiple SQL statements with named parameters in batch mode at once, if they are deposited in the following input XML structure, each within a `<namedpreparedstatementbatch>` block. In batch mode, only SQL statements can be used that do not return a `ResultSet`, i.e. no queries with `SELECT`.

Input

```
<any element name>
   <namedpreparedstatementbatch resulttag="result element name" timezone="time zone"
    format="date format">
      <sql>
         <!--SQL statement (optionally within a CDATA block) -->
         <!--Each parameter has an XML element with the same name as the
corresponding <row> child element -->
      </sql>
      <rowset>
         <row>
            <Name><!--value--></Name>
         </row>
         <!-- Further <row> elements for batch mode -->
      </rowset>
   </namedpreparedstatementbatch>
</any element name>
```

The following rules apply to each `<namedpreparedstatementbatch/>` element:

- The optional `timezone` attribute specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat` and defines the returned date values' formats.
- It contains at least one `<sql/>` element.

The following rules apply to each named parameter XML element within element `<sql/>`:

- The element name for the parameter is equivalent to the element name within the corresponding `<row/>` element.

- The optional attribute `mode` indicates the "direction" of the data stream (*`in`*: Input parameter for the database; *`out`*: Result parameter from the database; *`inout`*: Both, input parameter and result parameter; default: *`in`*.
- The optional attribute `type` specifies the data type, e. g. *`INTEGER`*, *`BLOB`*, or *`VARCHAR`* (default).
- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *`GMT+2`*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`. and defines the expected value format of the corresponding `<row>` element.

> If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ SELECT * FROM TABLE WHERE ID<]]><id type="INTEGER" />`.

The following rules apply to each `<rowset/>` element:

- It can be used as often as required (`0-n` times)
- It contains any number of `<row/>` elements controlling the statements' execution.

The following rules apply to each `<row/>` element:

- It can contain at least one elements with a name that corresponds to the parameter name in the SQL statement.
- The elements inside contain the corresponding value.

For each data element within `<row/>` make sure to avoid accidental line breaks within the element value, as all line breaks would be converted to `\n` in the generated SQL statement.

> **Please note**
> Allowed data types for the attribute `type` when setting (input parameter) and reading (*ResultSet*) values are: *`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`* and *`VARCHAR`* (default), *`XML`* (alias for SQLXML).

Example

```
<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

If the SQL statements were executed successfully, the adapter outputs the following XML document:

```
<Data>
   <result-element-name>
      <!--status of the batch operation-->
   </result-element-name>
</Data>
```

The following status outputs are possible for the batch operation:

- *(number of rows)* `rows affected` or
- `SUCCESS_NO_INFO` or
- `EXECUTE_FAILED`

10.6.1.10 Passing XML Structures

The *JDBC Connector* allows to pass XML structures. Within an input XML document you can define custom XML structures in addition to the SQL statement structures that are directly output by the adapter. This may simplify the process modeling in many cases by enabling to process those XML structures together with the SQL statement(s)' results.

Input

With the following statement defined as child of the root element in your input XML document, you can pass one or several XML structures:

```
<copy resulttag="name of the parent element">
  any XML structure
</copy>
```

Output

The XML structure defined within the `<copy>` element will be output by the *JDBC Connector* without Processing Instructions and comments. Thereby, the XML structure is enclosed by an element specified in the input structure within the attribute `resulttag`.

```
<name of the parent element>
  Input XML structure
</name of the parent element>
```

Example

Additionally to an SQL statement, an XML structure for the passing is defined in the following input XML document:

```
<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>
```

As a result, the *JDBC Connector* outputs e. g. the following result XML document containing the XML structure with the specified parent elements:

```
<Data>
  <result>
    <USERID>user</USERID>
    <PASSWD>password</PASSWD>
    <CLIENTID isNull="true" />
  </result>
  <Content>
    <Data>Text</Data>
  </Content>
</Data>
```

10.6.1.11 Retrieving Database Metadata

The parameter `AddMetaDataForQueries` (*X4 Server Revision 32408* or newer) enables to output database metadata to the respective result set in the result XML document, in common with the results of the SQL statement.

The name of each XML element containing the corresponding metadata consists of the value of the attribute `resulttag` in the input document and the appendix `-meta`, e. g. `<result-meta>`.

No metadata can be output for SQL statements that do not produce a result set (e. g. `UPDATE` or `DELETE`). When using the adapter's batch mode (`StatementBatch`, `PreparedStatementBatch` or `CallableStatementBatch`) this feature is also not available.

Example

The following input XML document defines an SQL query and a result element with the name `result`:

```
<Root>
   <statement resulttag="result">
      <sql> SELECT X4USER, STARTTIME FROM HISTORY WHERE PID='1279017294476' </sql>
   </statement>
</Root>
```

Then, the *JDBC Connector* outputs e. g. the following result XML document that contains the SQL query's results as well as the result set's metadata:

```
<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`: Connects to a JDBC-compatible database and executes one or multiple database statements handed over as input XML document with an adapter-specific structure. The iterator retrieves and provides the data for processing block by block. The database connection remains open until all data has been retrieved.

The connector requires a JDBC datasource that has been deployed within the application server and for which the connection URL, credentials, schema and transaction behavior etc. are defined. For further information on deploying JDBC datasources consult your application server's documentation.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *`Iterate`*: Read the amount of data from the database depending on the value specified within the parameter `fetchSize` and output it in a result document • *`Clean`*: Reset the iterator of a *JDBC Iterator*, which has the same ID as the desired *JDBC Iterator* with the operation *`Iterate`* (see parameter `id`)

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.jdbc.JdbcIterator*: Main class (Default)
JndiName	JNDI name of the DataSource If the connection is established via the JNDI name, the parameter `connectionURL` must not be specified. **Possible values:** Valid JNDI name (e. g. *`java:/DefaultDS`*)
ColumnCase	Controls how column names are taken over into the result. Since databases have different behaviors regarding the upper and lower case spelling of columns, the processing of the result can be specified here regardless of the database's behavior. **Possible values:** • *`NO_CHANGE`*: Does not change the column names, i. e. the column names will be written to the result document as returned by the database (Default) • *`TO_LOWER`*: Changes the column names to lower case • *`TO_UPPER`*: Changes the column names to upper case Column names, which are not legal element names (`NCNAME`) are converted to `_<number>` (e. g. `_2`) where *`<number>`* is the column index.

connectionURL	Connection URL for the JDBC database If the connection is established via the connection URL, the parameter `JndiName` must not be specified. **Possible values:** • The URLs structure depends on the database to which a connection is to be established. • Example for sqlite: *`jdbc:sqlite:C:/Temp/mydatabase.db`*
user	User name for authentication via the parameter `connectionURL`
password	Password for authentication via the parameter `connectionURL`
CommitMode	Defines the version to be used for commits **Possible values:** • *`NONE`*: Do not execute any commit • *`AUTO`*: Automatic commit according to the JDBC API (i. e. after each batch) (Default) • *`STATEMENT`*: Explicit commit after each statement (i. e. after all batches for this statement) • *`DOCUMENT`*: At the end of the document (i. e. after executing all statements) • *`MANAGED`*: Use adapter in transactions (for externally managed transactions)
AddMetaDataForQueries	Controls whether database metadata, see JDBC Metadata Explorer, should be output for each *result set* within the adapter's result XML document, if the adapter executes an SQL query (not available for `UPDATE`, `DELETE` etc.). This function is neither available in batch mode (`StatementBatch`, `PreparedStatementBatch`, and `CallableStatementBatch`). **Possible values:** • *`true`*: Add database metadata to result document • *`false`*: Output result document without database metadata (Default)
fetchSize	The number of results per query can be limited by entering a positive integer. If there is data available, the status `1` is returned, if the result is empty, the status `0` is output. If the entered number is smaller than `1`, the complete data will be fetched.

id	Unique ID of the *JDBC Iterator* component within the process context, respectively the process instance (if you want to use several *JDBC Iterators* that iterate regardless of each other within the same process context); **Possible values:** • *0*: Use current action ID (number of the *JDBC Iterator's* process step within the current process) as ID. Please note, that a *JDBC Iterator* may have the same action ID within a subprocess (default) • Any integer except *0*, in order to mark the corresponding *JDBC Iterator* component within the process context clearly. Recommendation: use a value *>1000* to avoid using an already existing action ID.

Status values

1	The query has returned data records or the instruction has changed data records The status *1* is returned, if the document contains multiple queries or instructions and at least one query has delivered or changed a result. The adapter returns this status as long as data blocks are still available for processing.
0	The query did not return any results or the statement hasn't changed any data records The status *0* is returned, if the document contains multiple queries or instructions and no query has returned or changed a result.
-1	Technical error: more information on the error cause can be found in the server log

Input

The *JDBC Iterator* excepts specific input XML structures containing database statements. These input XML structures can be generated dynamically by using XSL mappings that depend on the processed XML data.

10.6.2.1 Statement: Direct SQL

Input

```
<Any element name>
   <statement resulttag="result element name" timezone="time zone"
    format="date format">
      <sql>
         SQL statement
      </sql>
   </statement>
</Any element name>
```

The following rules apply to each `<statement/>` element:

- The optional attribute `timezone` specifies a timezone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`.
- It contains at least one `<sql/>` element with an SQL statement.

> If you use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]>`

Example

```
<select>
   <statement resulttag="result" timezone="UTC" format="dd.MM.yyyy">
      <sql>
         <![CDATA[SELECT * FROM JMS_USERS]]>
      </sql>
   </statement>
</select>
```

Output

If the SQL statement was executed successfully, the database connection component will output an XML document with the following structure containing the query response as a string:

```
<Data>
   <Result element name>
      <Feld1><!--Value--></Feld1>
      <Feldn><!--Value--></Feldn>
   </Result element name>
</Data>
```

> For `SELECT` statements used to access database tables with column names that do not represent a valid XML name, synthetic element names are created within the result XML document, in order to ensure that the XML document is well-formed.

10.6.2.2 StatementBatch: Direct SQL in Batch Mode

Input

In batch mode, the *JDBC Iterator* executes multiple SQL statements at the same time, if they are deposited in the following input XML structure. In this case, only SQL statements are possible, which do not return a `ResultSet`, i.e. no queries with `SELECT`:

```
<Any element name>
   <statementbatch resulttag="Result element name" timezone="time zone"
    format="date format">
      <sql>
         SQL statement 1
      </sql>
      <sql>
         SQL statement n
      </sql>
   </statementbatch>
</Any element name>
```

The following rules apply to each `<statementbatch/>` element:

- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`.
- It contains at least one `<sql/>` element that contains an SQL statement.

> If you want to use characters within the SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=100 ]]>`

Example

```
<select>
   <statementbatch resulttag="Result element name" 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

If the batch statement was executed successfully, the database connection element will output the following structure:

```
<Data>
   <Result element name>
      <!--Status of the batch operation-->
   </Result element name>
</Data>
```

The following status outputs are possible for the batch operation:

- *(Number or rows)* rows affected.
- SUCCESS_NO_INFO
- EXECUTE_FAILED

10.6.2.3 PreparedStatement: SQL with Values

The *JDBC Iterator* executes SQL statements with dynamical input data, if they are deposited in the following input XML structure.

> Alternatively, since X4 Suite release 5.0 you can execute SQL statements using named parameters, see NamedPreparedStatement: SQL with Named Parameters.

Input

```
<Any element name>
   <preparedstatement resulttag="Result element name" timezone="time zone"
    format="date format">
      <sql>
         <!--SQL statement-->
      </sql>
      <columns>
         <column mode="in" name="any NCName"
          index="placeholder index" type="data type"
          timezone="time zone" format="date format">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--value--></Name>
         </row>
      </rowset>
   </preparedstatement>
</Any element name>
```

The following rules apply to each <preparedstatement/> element:

- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`.
- It contains at least one `<sql/>` element.

> If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]>`

The following rules apply to each `<column/>` element:

- It can be used as often as required (`0-n` times)
- The optional attribute `mode` indicates the "direction" of the data stream (*in*: Input parameter for the database; *out*: Result parameter from the database; *inout*: Both, input parameter and result parameter; default: *in*.
- The attribute `name` specifies any NC name for a parameter.
- The attribute `index` specifies an integer placeholder index for the parameter defined within the attribute `name`.
- The optional attribute `type` specifies the data type (e. g. *INTEGER* or *BLOB; Default: VARCHAR*).
- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`.

The following rules apply to each `<rowset/>` element:

- It can be used as often as required (`0-n` times)
- It contains at least one element `<row/>` controlling the statements' execution.

The following rules apply to each `<row/>` element:

- It can contain any number of elements with the column name or the column alias set in the SQL statement.
- The elements inside contain the respective value.

Please note

- The attributes `mode` and `format` are required parameters up till X4 Server Revision 31469; in newer version they are optional.
- Allowed data types for the attribute `type` when setting (input parameter) and reading (*ResultSet*) values are: *`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`* (default), *`XML`* (alias for SQLXML).

Example

```
<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

If the SQL statement was executed successfully, the adapter outputs an XML document containing the query response as a string with the following structure:

```
<Data>
   <Result element name>
      <Field1><!--Value--></Field1>
      <Fieldn><!--Value--></Fieldn>
   </Result element name>
</Data>
```

10.6.2.4 PreparedStatementBatch: SQL with Values in Batch Mode

In batch mode, the *JDBC Iterator* executes multiple SQL statements at the same time, if they are deposited in the following input XML structure. In this case, only SQL statements are possible, which do not return a `ResultSet`, i.e. no queries with `SELECT`.

> ⓘ Alternatively, since X4 Suite release 5.0 you can execute SQL statements in batch mode using named parameters, see NamedPreparedStatementBatch: SQL with Named Parameters in Batch Mode.

Input

```
<Any element name>
   <preparedstatementbatch resulttag="Result element name" timezone="time zone"
    format="date format">
      <sql>
         <!-- SQL statement -->
      </sql>
      <columns>
         <column mode="in" name="any NCName"
          index="placeholder index" type="data type"
          timezone="time zone" format="data format">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--value--></Name>
         </row>
         <!-- Further <row> elements for batch mode -->
      </rowset>
   </preparedstatementbatch>
</Any element name>
```

The following rules apply to each `<preparedstatementbatch/>` element:

- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`.
- It contains at least one `<sql/>` element.

> ⓘ If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ UPDATE PROCESS SET X4_VERSION=4 WHERE X4_DURATION=?]]>`

The following rules apply to each `<column/>` element:

- It can be used as often as required (`0–n` times)
- The optional attribute `mode` indicates the "direction" of the data stream (*`in`*: Input parameter for the database; *`out`*: Result parameter from the database; *`inout`*: Both, input parameter and result parameter; default: *`in`*.
- The attribute `name` specifies any NC name for a parameter.
- The attribute `index` specifies an integer placeholder index for the parameter defined within the attribute `name`.
- The optional attribute `type` specifies the data type (e. g. *`INTEGER`* or *`BLOB; Default: VARCHAR`*).
- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *`GMT+2`*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat`.

The following rules apply to each `<rowset/>` element:

- It can be used as often as required (`0–n` times)
- It contains any number of `<row/>` elements controlling the statements' execution.

The following rules apply to each `<row/>` element:

- It can contain any number of elements with the column name or the column alias set in the SQL statement.
- The elements inside contain the respective value.

Please note

- The attributes `mode` and `format` are required parameters up till X4 Server Revision 31469; in newer version they are optional.
- Allowed data types for the attribute `type` when setting (input parameter) and reading (*ResultSet*) values are: *`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 and VARCHAR`* (default), *`XML`* (alias for SQLXML).

Example

```
<Rootelement>
   <preparedstatementbatch resulttag="Result element name" 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

If the batch statement was executed successfully, the database connection component outputs the following XML structure:

```
<Data>
   <Result element name>
      <!--status of the batch operation-->
   </Result element name>
</Data>
```

The following status outputs are possible for the batch operation:

- *(number of rows)* rows affected
- SUCCESS_NO_INFO
- EXECUTE_FAILED

10.6.2.5 CallableStatement: Stored Procedures

The *JDBC Iterator* executes stored procedures via *JDBC callable statements* with dynamical input data, if they are deposited in the following input XML structure.

Input

```
<any element name>
   <callablestatement resulttag="result element name" timezone="timezone"
    format="date format">
      <sql>
        <!--SQL statement -->
      </sql>
      <columns>
         <column mode="mode" name="any name" index="placeholder"
          type="data type" timezone="time zone" format="date format">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--value--></Name>
         </row>
      </rowset>
   </callablestatement>
</any element name>
```

The following rules apply to each `<callablestatement/>` element:

- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*; see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a data format according to `java.text.SimpleDateFormat`.
- It contains at least one `<sql/>` element.

> If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]>`

The following rules apply to each `<column/>` element:

- It can be used as often as required (`0-n` times)
- The optional attribute `mode` indicates the "direction" of the data stream (*in*: Input parameter for the database; *out*: Result parameter from the database; `inout`: Both input parameter and result parameter; see http://download.oracle.com/javase/1,5.0/docs/guide/jdbc/getstart/callablestatement.html); Default: *in*.
- The attribute `name` specifies an arbitrary name for a parameter.
- The attribute `index` specifies an integer placeholder index for the parameter defined within the attribute `name`.
- The optional attribute `type` specifies the data type (allowed data types see note below; default: *VARCHAR*).
- The attribute `name` specifies an arbitrary name for a parameter.
- The optionale Attribut `timezone` specifies a time zone according to `java.util.TimeZone`, e.g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.

- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat` an.

The following rules apply to each `<rowset/>` element:

- It can be used as often as required (*0–n* times).
- It contains any number of `<row/>` elements controlling the statements' execution.

The following rules apply to each `<row/>` element:

- It can contain any number of elements with the column name or the column alias set in the SQL statement.
- The elements inside contain the respective value.

Please note

- The attributes `mode` and `format` are required parameters up till X4 Server Revision 31469; in newer version they are optional.
- Allowed data types for the attribute `type` when setting (input parameter) and reading (*ResultSet*) values are: *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* and *VARCHAR*(default), *XML* (alias for SQLXML).

Example

```
<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

If the SQL statement was executed successfully, the database connection element outputs an XML document containing the query response as a string with the following structure:

```
<Data>
   <<result element name>
      <Field1><!--value--></Field1>
      <Fieldn><!--value--></Fieldn>
   </<result element name>
</Data>
```

10.6.2.6 CallableStatementBatch: Stored Procedures in Batch Mode

The *JDBC Iterator* executes stored procedures via *JDBC callable statements* with dynamical input data, if they are specified with the following input XML structure. Only SQL statements are feasible that do not return any `ResultSet`, for example no queries with a `SELECT`.

Input

```
<any element name>
   <callablestatementbatch resulttag="result element name" timezone="timezone"
 format="date format">
      <sql>
        <!--SQL statement -->
      </sql>
      <columns>
         <column mode="mode" name="any name" index="placeholder" type="data type"
 timezone="timezone" format="date format">
         </column>
      </columns>
      <rowset>
         <row>
            <Name><!--value--></Name>
         </row>
      </rowset>
   </callablestatementbatch>
</any element name>
```

The following rules apply to each `<callablestatementbatch/>` element:

- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*; see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a data format according to `java.text.SimpleDateFormat`.
- It contains at least one `<sql/>` element.

> If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ SELECT * FROM TABLE WHERE ID<1 ]]`

The following rules apply to each `<column/>` element:

- It can be used as often as required (`0–n` times)
- The optional attribute `mode` indicates the "direction" of the data stream (*`in`*: Input parameter for the database; *`out`*: Result parameter from the database; `inout`: Both input parameter and result parameter; see http://download.oracle.com/javase/1,5.0/docs/guide/jdbc/getstart/callablestatement.html); Default: *`in`*.
- The attribute `name` specifies an arbitrary name for a parameter.
- The attribute `index` specifies an integer placeholder index for the parameter defined within the attribute `name`.
- The optional attribute `type` specifies the data type (allowed data types see note below; default: *`VARCHAR`*).
- The attribute `name` specifies an arbitrary name for a parameter.
- The optionale Attribut `timezone` specifies a time zone according to `java.util.TimeZone`, e.g. *`GMT+2`*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat` an.

The following rules apply to each `<rowset/>` element:

- It can be used as often as required (*`0–n`* times).
- It contains any number of `<row/>` elements controlling the statements' execution.

The following rules apply to each `<row/>` element:

- It can contain any number of elements with the column name or the column alias set in the SQL statement.
- The elements inside contain the respective value.

Please note

- The attributes `mode` and `format` are required parameters up till X4 Server Revision 31469; in newer version they are optional.
- Allowed data types for the attribute `type` when setting (input parameter) and reading (*ResultSet*) values are: *`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`* and *`VARCHAR`* (default), *`XML`* (alias for SQLXML).

Example

```
<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

If the batch statement was executed successfully, the database connection element outputs the following XML structure:

```
<Data>
   <result element name>
      <!--status of the batch operation-->
   </result element name>
</Data>
```

The following status outputs of the batch operation are possible:

- *(number of rows)* `rows affected.` or
- `SUCCESS_NO_INFO` or
- `EXECUTE_FAILED`

10.6.2.7 NamedPreparedStatement: SQL with Named Parameters

The *JDBC Iterator* executes SQL statements with named parameters, if they are deposited in the following input XML structure.

Input

```
<any element name>
   <namedpreparedstatement resulttag="result element name" timezone="time zone"
    format="date format">
      <sql>
         <!--SQL statement (optionally within a CDATA block) -->
         <!--Each parameter has an XML element with the same name as the
corresponding <row> child element -->
      </sql>
      <rowset>
         <row>
            <Name><!--value--></Name>
         </row>
      </rowset>
   </namedpreparedstatement>
</any element name>
```

The following rules apply to each `<namedpreparedstatement/>` element:

- The optional `timezone`attribute specifies a time zone according to `java.util.TimeZone`, e. g. *`GMT+2`*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat` and defines the returned date values' formats.
- It contains one `<sql/>` element.

The following rules apply to each named parameter XML element within element `<sql/>`:

- The element name for the parameter is equivalent to the element name within the corresponding `<row/>` element.
- The optional attribute `mode` indicates the "direction" of the data stream (*`in`*: Input parameter for the database; *`out`*: Result parameter from the database; *`inout`*: Both, input parameter and result parameter; default: *`in`*.
- The optional attribute `type` specifies the data type, e. g. *`INTEGER`*, *`BLOB`*, or *`VARCHAR`*(default).
- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *`GMT+2`*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat` and defines the expected value format of the corresponding `<row>` element.

If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ SELECT * FROM TABLE WHERE ID<]]><id type="INTEGER" />`.

The following rules apply to each `<rowset/>` element:

- It can be used as often as required (`0-n` times)
- It contains any number of `<row/>` elements controlling the statements' execution.

The following rules apply to each `<row/>` element:

- It must contain at least one element corresponds to the element name within the SQL statement.
- The elements inside contain the corresponding values.

Make sure to avoid unintended line breaks within the element content of each data element within `<row/>`, as all line breaks would be converted to `\n` in the generated SQL statement.

Please note
Allowed data types for the attribute `type` when setting (input parameter) and reading (*ResultSet*) values are: *`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` (default), *`XML`* (alias for SQLXML).

Example

```
<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

If the SQL statement was executed successfully, the database connection element outputs an XML document containing the query response as a string with the following structure:

```
<Data>
   <result element name>
      <Field1><!--value--></Field1>
      <Fieldn><!--value--></Fieldn>
   </result element name>
</Data>
```

10.6.2.8 NamedPreparedStatementBatch: SQL with Named Parameters in Batch Mode

The *JDBC Iterator* executes multiple SQL statements in batch mode at once, if they are deposited in the following input XML structure, each within a `<namedpreparedstatementbatch>` block. Thereby, only SQL statements can be used that do not return a `ResultSet`, i.e. no queries with `SELECT`.

Input

```
<any element name>
   <namedpreparedstatementbatch resulttag="result element name" timezone="time zone"
    format="date format">
      <sql>
         <!--SQL statement (optionally within a CDATA block) -->
         <!--Each parameter has an XML element with the same name as the
corresponding <row> child element -->
      </sql>
      <rowset>
         <row>
            <Name><!--value--></Name>
         </row>
         <!-- Further <row> elements for batch mode -->
      </rowset>
   </namedpreparedstatementbatch>
</any element name>
```

The following rules apply to each `<namedpreparedstatementbatch/>` element:

- The optional `timezone` attribute specifies a time zone according to `java.util.TimeZone`, e. g. *GMT+2*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat` and defines the returned date values' formats.
- It contains at least one `<sql/>` element.

The following rules apply to each named parameter XML element within element `<sql/>`:

- The element name for the parameter is equivalent to the element name within the corresponding `<row/>` element.

- The optional attribute `mode` indicates the "direction" of the data stream (*`in`*: Input parameter for the database; *`out`*: Result parameter from the database; *`inout`*: Both, input parameter and result parameter; default: *`in`*.
- The optional attribute `type` specifies the data type, e. g. *`INTEGER`*, *`BLOB`*, or *`VARCHAR`* (default).
- The optional attribute `timezone` specifies a time zone according to `java.util.TimeZone`, e. g. *`GMT+2`*, see http://docs.oracle.com/javase/1.5.0/docs/api/java/util/TimeZone.html.
- The optional attribute `format` specifies a date format according to `java.text.SimpleDateFormat` and defines the expected value format of the corresponding `<row>` element.

> If you want to use characters within SQL statements that have a meaning in XML, escape the SQL statement with a surrounding `CDATA` block, e. g. `<![CDATA[ SELECT * FROM TABLE WHERE ID<]]><id type="INTEGER" />`.

The following rules apply to each `<rowset/>` element:

- It can be used as often as required (`0-n` times)
- It contains any number of `<row/>` elements controlling the statements' execution.

The following rules apply to each `<row/>` element:

- It must contain at least one element corresponding to the element name in the SQL statement.
- The elements inside contain the corresponding value.

Make sure to avoid unintended line breaks within the element content of each data element within `<row/>`, as all line breaks would be converted to `\n` in the generated SQL statement.

> **Please note**
> Allowed data types for the attribute `type` when setting (input parameter) and reading (*ResultSet*) values are: *`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`* and *`VARCHAR`* (default), *`XML`* (alias for SQLXML).

Example

```
<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
If the SQL statements were executed successfully, the adapter outputs the following XML document:

```
<Data>
   <result-element-name>
      <!--status of the batch operation-->
   </result-element-name>
</Data>
```

The following status outputs are possible for the batch operation:

- *(number of rows)* `rows affected` or
- `SUCCESS_NO_INFO` or
- `EXECUTE_FAILED`

10.6.2.9 Retrieving Database Metadata

The parameter `AddMetaDataForQueries` (since *X4 Server Revision 32408*) enables to output database metadata to the respective result set in the result XML document, in common with the results of the SQL statement.

The name of each XML element containing the corresponding metadata consists of the `resulttag` attribute's value within the input document and the appendix `-meta`, e. g. `<result-meta>`.

> For SQL statements that do not produce a result set (e. g. `UPDATE` or `DELETE`) no metadata can be output. When using the adapter's batch mode (`StatementBatch`, `PreparedStatementBatch` or `CallableStatementBatch`) this feature is neither available.

Example

The following input XML document defines an SQL query and a result element with the name `result`:

```
<Root>
   <statement resulttag="result">
      <sql> SELECT X4USER, STARTTIME FROM HISTORY WHERE PID='1279017294476' </sql>
   </statement>
</Root>
```

Then, the *JDBC Iterator* outputs e. g. the following result XML document that contains the SQL query's results as well as the result set's metadata:

```
<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`: Establishes a connection to a JDBC-compatible database and determines metadata about the database, tables and columns as well as stored primary and foreign keys. The adapter expects an input XML document with adapter-specific structures to query the desired metadata.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *execute*: execute adapter

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.jdbc.meta.MetadataExplorer`*: Main class (default)
JndiName	JNDI name of the DataSource If the connection is established via the JNDI name, the parameter `connectionURL` must not be filled in. **Possible values:** Valid JNDI name (e.g. *`java:/DefaultDS`*)
ColumnCase	Controls how the column names are taken over into the result. Since databases have different behaviour regarding the upper/lower case of column names, this can be used to make the further processing of the result independent of the database behaviour. **Possible values:** • *`NO_CHANGE`*: Does not change the column names, i.e. the spelling in the result is exactly as supplied by the database (default). • *`TO_LOWER`*: Changes the column names to lower case. • *`TO_UPPER`*: Changes the column names to capital letters Column names that are not valid element names (`NCNAME`) are replaced by `_<number>` (e.g. `_2`), where *`<number>`* is the index of the column.
connectionURL	Connection URL for the JDBC database If the connection is established via the connection URL, the parameter `JndiName` must not be filled in. **Possible values:** • The type of setup depends on the database to which a connection is to be established. • Example for sqlite: *`jdbc:sqlite:C:/Temp/mydatabase.db`*
user	User name for authentication when using the parameter `connectionURL`
password	Password for authentication when using the parameter `connectionURL`

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.

`-1`	The operation failed due to a technical error.

Input

The *JDBC Metadata* Explorer expects specific input XML structures, which can contain one or more of the metadata queries described below, as required. These input XML structures can be generated dynamically via XSL mappings depending on the processed XML data.

- Retrieving database metadata
- Retrieving table metadata
- Retrieving column metadata
- Retrieving primary keys of a table
- Retrieving foreign keys of a table
- Retrieving referenced foreign keys

10.6.3.1 Retrieving database metadata

The *JDBC Metadata Explorer* enables the retrieval of different database metadata (vendor/product name, supported functions, etc.) and the retrieval of existing schemas from the database that is connected via JDBC.

Input

Add the element `<databasemeta resulttag="`*Result element name*`" />` to an input XML document with any root element, and specify a valid XML name within its attribute `resulttag`.

```
<Root>
  <databasemeta resulttag="Result element name"/>
</Root>
```

Output

The output XML structure contains the root element `<Data>` with a child element, whose name has been specified in the input structure within the attribute `resulttag`. It contains schema information and database metadata:

```
<Data>
   <ResultElementName key="value" ...>
   <Schema>
      <TABLE_SCHEM>DB-Schema</TABLE_SCHEM>
      <TABLE_CATALOG isNull="(true/false)" />
   </Schema>
   ...
   </ResultElementName>
</Data>
```

Example

An input XML document contains the following XML structure to output metadata for the connected database:

```
<RootElement>
   <databasemeta resulttag="DB"/>
</RootElement>
```

Then, the *JDBC Metadata Explorer* outputs e. g. the following result XML document:

```
<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 Retrieving table metadata

The *JDBC Metadata Explorer* enables the retrieval of different database table metadata (name, type etc.); you can narrow the search by specifying catalog, schema, and table name.**Input**

Add the element `<tablemetadata resulttag="`*Result element name*`"/>` to an input XML document with any root element, and specify a valid XML name within the attribute `resulttag`. The specification of catalog, schema, and table within the elements `<catalog>`, `<schema>`, and `<table>` are optional here; if these are not specified, all available metadata will be retrieved.

```
<Root>
  <tablemetadata resulttag="ResultElementName">
     <catalog>Catalog</catalog>
     <schema>Database schema</schema>
     <table>Database table</table>
  </tablemetadata>
</Root>
```

Output

The output XML structure contains the root element `<Data>` with a child element whose name has been specified in the input structure within the attribute `resulttag`. It contains table metadata. If there are multiple result sets (i. e. more than one table or view), each will be output in a separate element.

```
<Data>
   <ResultElementName>
      <Table property> value </Table property>
      ...
   </ResultElementName>
</Data>
```

Further information on each table property can be found here.

Example

An input XML document contains the following XML structure to output metadata for a table HISTORY in a database schema APP:

```
<Root>
  <tablemetadata resulttag="DBTable">
     <catalog></catalog>
     <schema>APP</schema>
     <table>HISTORY</table>
  </tablemetadata>
</Root>
```

Then, the *JDBC Metadata Explorer* outputs e. g. the following result XML document:

```
<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 Retrieving column metadata

The *JDBC Metadata Explorer* enables the retrieval of different column metadata (data type, nullable, autoincrement etc.) to one or multiple table columns; you can narrow the search by specifying catalog, schema, and table name.

Input

Add the element `<tablecolumns resulttag="ResultElementName"/>` to an input XML document with any root element, and specify a valid XML name within the attribute `resulttag`. The specification of catalog, schema, table and column within the elements `<catalog>`, `<schema>`, `<table>` and `<column>` are optional here; if these are not specified, all available metadata will be retrieved.

```
<Root>
  <tablemetadata resulttag="ResultElementName">
     <catalog>Katalog</catalog>
     <schema>DB-Schema</schema>
     <table>DB-Tabelle</table>
     <column>Spalte</column>
  </tablemetadata>
</Root>
```

Output

The output XML structure contains the root element `<Data>` with a child element whose name has been specified in the input structure within the attribute `resulttag`. It contains column metadata. If there are multiple result sets (i. e. more than one column), each will be output in a separate element.

```
<Data>
   <ResultElementName>
      <Column property> value </Column property>
      ...
   </ResultElementName>
</Data>
```

Further information on each table property can be found here.

Example 1

An input XML document contains the following XML structure to output metadata for a column `X4_PID` in the table `HISTORY` in a database schema APP:

```
<Root>
  <tablecolumns resulttag="DBColumn">
     <catalog></catalog>
     <schema>APP</schema>
     <table>HISTORY</table>
     <column>X4_PID</column>
  </tablecolumns>
</Root>
```

Then, the *JDBC Metadata Explorer* outputs e. g. the following result XML document:

```
<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>
```

Example 2

An input XML document contains the following XML structure to output metadata for a column `X4_PID` in the table `HISTORY` in a database schema APP. The element `<column>` with its attribute `label` within `<resultsetRestriction>` restricts the output data::

```
<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>
```

The *JDBC Metadata Explorer* outputs e. g. the following result XML document:

```
<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 Retrieving primary keys of a table

The *JDBC Metadata Explorer* enables the retrieval of primary keys (name and column information) of a specified database table.

Input

Add the element `<primarykeys resulttag="`*ResultElementName*`"/>` to an input XML document with any root element, and specify a valid XML name within the attribute `resulttag`. It is required to specify a table in element `<table>`; the elements `<catalog>` and `<schema>` are optional here.

```
<Root>
  <primarykeys resulttag="ResultElementName">
     <catalog>Catalog</catalog>
     <schema>Database schema</schema>
     <table>Database table (mandatory, may not be empty)</table>
  </primarykeys>
</Root>
```

Output

The output XML structure contains a root element `<Data>` with a child element whose name has been specified in the input structure within the attribute `resulttag`. It contains metadata to each primary key of the table. If there are multiple result sets (i. e. more than one primary key column), each will be output in a separate element.

```
<Data>
   <ResultElementName>
      <PrimayKeyProperty> value </PrimayKeyProperty>
      ...
   </ResultElementName>
</Data>
```

Further information on each column property can be found here.

Example

An input XML document contains the following XML structure to output the primary keys of table `HISTORY`:

```
<Root>
   <primarykeys resulttag="historykeys">
      <schema></schema>
      <table>HISTORY</table>
   </primarykeys>
</Root>
```

Then, the *JDBC Metadata Explorer* outputs e. g. the following result XML document:

```
<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 Retrieving foreign keys of a table

The *JDBC Metadata Explorer* enables the retrieval of foreign keys (name and column information) of a specified database table.

Input

Add the element `<foreignkeys resulttag="`*ResultElementName*`"/>` to an input XML document with any root element, and specify a valid XML name within the attribute `resulttag`. It is required to specify a table in element `<table>`; the elements `<catalog>` and `<schema>` are optional here.

```
<Root>
  <foreignkeys resulttag="ResultElementName">
     <catalog>Catalog</catalog>
     <schema>Database schema</schema>
     <table>Database table (mandatory, may not be empty)</table>
   </foreignkeys>
</Root>
```

Output

The output XML structure contains a root element `<Data>` with a child element whose name has been specified in the input structure within the attribute `resulttag`. It contains metadata to each foreign key of the table. If there are multiple result sets (i. e. more than one foreign key column), each will be output in a separate element.

```
<Data>
   <ResultElementName>
      <ForeignKeyProperty> Value </ForeignKeyProperty>
      ...
   </ResultElementName>
</Data>
```

Further information on each column property can be found here.

Example

An input XML document contains the following XML structure to output the foreign keys of table Provider:

```
<Root>
   <foreignkeys resulttag="historykeys">
      <table>Provider</table>
   </foreignkeys>
</Root>
```

Then, the *JDBC Metadata Explorer* outputs e. g. the following result XML document:

```
<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 Retrieving referenced foreign keys

The *JDBC Metadata Explorer* enables the retrieval of foreign keys in other tables that reference to the specified database table.

Input

Add the element `<exportedkeys resulttag="ResultElementName"/>` to an input XML document with any root element, and specify a valid XML name within the attribute `resulttag`. It is required to specify a table within the element `<table>`; the elements `<catalog>` and `<schema>` are optional here.

```
<Root>
  <exportedkeys resulttag="ResultElementName">
     <catalog>Catalog</catalog>
     <schema>Database schema</schema>
     <table>Database table (mandatory, may not be empty)</table>
   </exportedkeys>
</Root>
```

Output

The output XML structure contains a root element `<Data>` with a child element whose name has been specified in the input structure within the attribute `resulttag`. It contains metadata to each foreign key in other tables that reference to primary key of the specified table. If there are multiple result sets (i. e. more than one foreign key column), each will be output in a separate element.

```
<Data>
   <ResultElementName>
      <ForeignKeyProperty> Value </ForeignKeyProperty>
      ...
   </ResultElementName>
</Data>
```

Further information on each column property can be found here.

Example

An input XML document contains the following XML structure to output the foreign key references of the table `Provider`:

```
<Root>
   <exportedkeys resulttag="keys">
      <table>Operator</table>
   </exportedkeys>
</Root>
```

Then, the *JDBC Metadata Explorer* outputs e. g. the following result XML document:

```
<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
- Supported EDIFACT Message Types

10.7.1 EDIFACT Adapter

? Unknown Attachment `EDIFACT Adapter`: Reads an EDIFACT message (raw data and Stage 1-3, *DIN ISO 9735* compliant) and outputs it as an UTF-8 encoded XML document. In addition, the adapter can output an EDIFACT message.

Former designation up to version 4.5.1: *EDIFACT Message Adapter*

Former name until version 5.5.4: *EDIFACT Converter*

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *`ReadRaw`*: Output EDIFACT message 1:1 as XML • *`ReadStage1`*: Read EDIFACT message as XML with concrete element names (*Stage 1*) • *`ReadStage2`*: Read EDIFACT message as XML in which the segments are organised based on the messages (*Stage 2*) • *`ReadStage3`*: Read EDIFACT message as XML in which the segments are structured with the correct hierarchy including nested segment groups in the tree, provided the adapter supports the message type (*Stage 3*) • *`Write`*: Write EDIFACT message

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.edifact.EdifactAdapter`*: Main class (default)

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without a result (input is *null*).
`-1`	The operation failed due to a technical error.

Input

An EDIFACT message of the type INVOIC (invoice) could be structured as follows:

```
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'
```

Output

With Operation `ReadRaw`, EDIFACT messages are output 1:1 as XML, e.g.:

```
<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>
```

Operation `ReadStage1` outputs EDIFACT messages as XML documents that already use concrete names instead of generic element names, e.g.:

```
<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>
```

With Operation `ReadStage2`, EDIFACT messages are output as XML documents in which the segments are organised based on the messages, e.g.:

```
<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>
```

With Operation `ReadStage3`, EDIFACT messages are output as XML documents in which the segments are structured with the correct hierarchy including nested segment groups in the tree, provided the adapter supports the message type, e.g.:

```
<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>
```

See also

- Adapter
- Supported EDIFACT Message Types

10.7.2 Supported EDIFACT Message Types

In X4 BPMS, a number of EDIFACT messages can be processed using the EDIFACT Adapter. The supported message types are constantly extended.
Currenty, the following message types and revisions are supported:

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 ?? (except IFCSUM*D*93A*UN) • IFCSUM Rev 3 (except IFCSUM*D*95A*UN - IFCSUM*D*95B*UN) • IFCSUM Rev 4 (except 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 (except IFTMBF*D*94A*UN) • IFTMBF Rev 2 (except IFTMBF*D*94B*UN) • IFTMBF Rev 3: IFTMBF*D*95A*UN • IFTMBF Rev 4 (except 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 (except IFTMCS*D*94A*UN) • IFTMCS Rev 2 (except IFTMCS*D*94B*UN) • IFTMCS Rev 3: IFTMCS*D*95A*UN • FTMCS Rev 4 (except 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 (except 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`: Receives, copies/moves or deletes an e-mail message in an Internet Message Access Protocol (IMAP) mailbox, sets a flag or lists message headers in a folder of an IMAP mailbox.

The adapter supports S/MIME encrypted/signed messages and SSL encryption of the connection to the IMAP server (*IMAPS*). Any *Java Mail properties* can be processed from a properties file.

Former name until version 4.5.1: *IMAP Connection Adapter*

Former name until version 5.5.4: *IMAP Connector*

The default block size (16 KB) during message transfer can be set via the JavaMail system property `mail.imap.fetchsize` in the runtime environment of the X4 server (usually Java VM with application server) to optimize the download of e-mails via IMAP.
Example of *JBoss* startup script, block size is set to 5 MB:

```
@set JAVA_OPTS=%JAVA_OPTS% -Dmail.imap.fetchsize=5242880
```

Properties

`Operation`	Determines which operation the adapter executes **Possible values:** • *`Receive Message`*: Receive message with ID specified in parameter `messageID` as UTF-8 encoded XML document. • *`Receive EML Message`*: Receive message with ID specified in parameter messageID and save as `.eml` file (MS Outlook) ⓘ You can forward the received message by directly connecting the POP/SMTP adapter after the IMAP adapter. • *`List Messages`*: List message headers of an IMAP folder specified in parameter `folder` in an UTF-8 encoded XML document. • *`Copy Message`*: Copy message whose message ID is specified in parameter `messageID` to IMAP folder specified in parameter `folder` • *`Move Message`*: Move message whose message ID is specified in parameter `messageID` to the IMAP folder specified in parameter `folder` • *`Delete Message`*: Delete message whose message ID is specified in parameter `messageID` from IMAP server • *`Set Flag`*: Mark message whose `messageID` with one or more `flags` specified in parameter flags

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.imap.Adapter*: Main class (default)
`imapServer`	IP address or host name of the IMAP server **Possible values:** Any valid string (e. g. *imap.1und1.de*)
`login`	Username for the email account used **Possible values:** Any string
`password`	Password for the email account used **Possible values:** Any string
`oAuth2`	Defines if the OAuth 2.0 protocol is used for log-in. **Possible values:** • *true*: oAuth2 is used for log-in • *false*: oAuth2 is not used for log-in ⚠ If the OAuth 2.0 protocol is used, • the SSL encryption must be enabled for the connection to the IMAP server by setting the `ssl` parameter. • the access token must be entered as the `password` parameter.
`folder`	Destination folder on the IMAP server (if the Operation property is set to *Copy Message* or *Move Message* gesetzt ist) or folder whose messages are to be listed (if the Operation property is set to *List Messages*) **Possible values:** Existing folder name (case sensitive) Frequently used: • INBOX • Drafts • Sent Messages • Junk • Deleted Messages • Archive • Notes

`srcFolder`	Source folder for *Copy Message*, *Move Message*, *Receive Message*, *Set Flag* and *Delete Message*. **Possible values:** Existing folder name (case sensitive) Frequently used: • INBOX • Drafts • Sent Messages • Junk • Deleted Messages • Archive • Notes
`flags`	IMAP flag(s) to identify an e-mail message; multiple flags are separated by semicolons (*;*) (if the Operation property is set to *Set Flag*) **Possible values:** Any flag (e.g. *\Answered*)
`messageID`	Message ID of the email including folder specification **Possible values:** folder name and message ID (with angle brackets) (e. g. *INBOX/<831921101.31255094008434.JavaMail.javamailuser@localhost>*)
`uid`	Message ID on the IMAP server **Possible values:** Any integer
`maxMessages`	Limit the number of messages to list when the operation property is set to *List Messages* gesetzt ist. The listing starts with the oldest messages in the folder. **Possible values:** Any integer
`base64`	Always base64 encode message attachments (multipart parts) (if Operation property is set to *Receive Message*) **Possible values:** • *yes*: Base64 encode all multipart parts • *no*: Do not base64 encode (default)

`encrypt`	Messages are encrypted with S/MIME (*Multipart/Encrypted*) **Possible values:** • *yes*: enable S/MIME encryption (requires specification of the key/certificate file in parameter Pkcs12 and the passphrase in parameter passphrase) • *no*: No encryption (default)
`sign`	Messages are signed with S/MIME (*Multipart/Signed*) **Possible values:** • *yes*: Process S/MIME signature (requires specification of key/certificate file in parameter Pkcs12 and passphrase in parameter passphrase) • *no*: No signature (default)
`Pkcs12`	Path to a valid key/certificate file in *PKCS#12* format **Possible values:** Any valid path specification in the local file system of the *X4 server* (e.g. *%USER_COLLECTION\Project\Folder\Key.p12*)
`passphrase`	Password/passphrase to the key/certificate file in *PKCS#12* format that you specified in parameter Pkcs12. **Possible values:** Any valid string
`port`	Port of the IMAP server **Possible values:** • Any integer (e.g. *993*) • *(no value)*: Use default port *143*
`ssl`	Enable SSL encryption for the connection to the IMAP server **Possible values:** • *Yes*: Use IMAP over SSL (IMAPS) • *No*: Do not encrypt connection (default)

`propertiesURL`	URL to a properties file (XML or text) that can contain any *Java Mail properties* **Possible values:** • Any valid URL to a properties file (e.g. *xstore:///Project/Folder/ mail.properties*) • *(empty)*: Do not *Java Mail properties*
`propertiesType`	Format of the properties file specified in `propertiesURL` parameter **Possible values:** • *TEXT*: Interpret as text properties (default) • *XML*: Interpret as XML properties

Status values

`1`	The operation was executed successfully.
`0`	The operation was performed, but no result was found or no data was changed.
`-1`	The operation could not be performed.

10.8.2 MIME Multipart

`MIME Multipart`: Allows to read and generate MIME multipart documents – primarily for the HTTP communication (e.g. ReSTful webservices). The adapter can also be used for the e-mail communication and for the handling of e-mail attachments.

The adapter supports you when working with MIME multipart documents. It converts MIME documents to an XML representation and the XML representation into MIME documents.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *`FromBytesWithHeaders:`* create a MIME XML document from a MIME document • *`FromRestMetaXml`* : create a corresponding MIME XML document from a ReST input XML • *`ModifyRestMetaXml`* : create a modified ReST input XML from a ReST input XML • *`ToRestMetaXml`* : create a ReST output XML with HTTP status code `200` from a MIME XML document. • *`ToBytesWithHeaders`* : create a MIME document from a MIME XML document • *`FromBytesWithBoundaryFromFirstLine`* : create a MIME XML document from a MIME body • *`ToBytesWithBoundaryInFirstLine`* : create a MIME body from a MIME XML document without preamble and without epilogue • *`FlattenXml`* : create a flat MIME XML document from a MIME XML document (if necessary multipart tree is serialized as single part) • *`PartFromX4Document:`* convert any incoming X4 document into a part in the Mime XML structure

Parameters

Adapter	Main adapter class (Do not change!) **Possible values:** *de.softproject.integration.adapter.mimemultipart.Adapter*: Main class (default)

10.8.2.1 Operation FromBytesWithHeaders

The *`FromBytesWithHeaders`* adapter operation can be used to process the components of a MIME document (e.g. an e-mail in EML format), e.g. the attachments of the e-mail.

Input: Example of a MIME XML document

```
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: Example of a generated MIME XML document

```
<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>
```

If the input document contains only a MIME document, only the corresponding part in a root `<part/>` element is output in the output XML document.

10.8.2.2 Operation FromRestMetaXml

The `FromRestMetaXml` adapter operation can be used to process XML documents in *MetaXml* format and output them as MIME XML. This input XML format is generated if an X4 process was called via the *X4 ReST Starter* and this process uses the *`HTTP-MetaXml`* input handler.

Examples of use:

- File upload using an HTML form: if data with the `multipart/form-data` content type is to be processed.
- Web services communication where all (meta) data of the HTTP request is relevant for processing.

Input: Example of an X4 ReST input document

```
<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: Example of a generated MIME (multipart) XML document.

```
<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

The `ModifyRestMetaXml` adapter operation can be used to process *MetaXml* documents generated by the *X4 ReST Starter* in the same way as the `FromRestMetaXml` operation, but the output document has a modified structure:

- The header area (`/Request/Headers`) is removed.
- The body element (`/Request/Body`) is replaced by the MIME XML structure.

Input: Rest input XML document, see Input for FromRestMetaXml operation).

Output: Example of a modified X4 ReST input document

```
<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

The `ToRestMetaXml` adapter operation can be used to read a MIME XML document and generate it in *MetaXML* format. This XML format is typically output as the result of an X4 process called via the *X4 ReST Starter* if it uses the *`HTTP-MetaXml`* output handler.

Input: Example of an X4 ReST input document

```
<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: Example of a generated X4 ReST output document.

```
<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

The adapter operation `ToBytesWithHeaders` can be used to create a MIME document with all required header information, i.e. all headers must be created by the user.

> ⚠ If the `Content` element has the attribute `encoding="base64"` set, the attachment must be Base64 encoded twice.

Use case: Creating an HTML e-mail with embedded images

Input: Example of a MIME XML document

```
<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: Example of a generated MIME document

```
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

The `FromBytesWithBoundaryFromFirstLine` adapter operation can be used to process parts of a MIME document (e.g. an e-mail in EML format) in an output XML document in the same way as the `FromBytesWithHeaders` adapter operation , but no header block is expected here and the MIME body is processed directly - provided it is a valid multipart boundary. Otherwise the input is output as a part without header in the generated XML document.

Use case: SOAP via JMS. Here, a byte message is used that contains a MIME body without a MIME header. To be able to use MTOM here as well, a MIME multipart body without preamble is used.

Input: Example of a MIME body

```
--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: Example of a generated MIME XML document

```
<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

The `ToBytesWithBoundaryInFirstLine` adapter operation can be used to create a MIME document without a header block and, in the case of multiparts, without a preamble and without an epilog.

Use case: SOAP via JMS (with MTOM)

Input: Example of a MIME XML document

```
<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: Example of a generated MIME body document.

```
--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

The `FlattenXml` adapter operation can be used to create a MIME multipart XML document that is prepared for sending with an X4 BPMS managed file transfer adapter, such as the *HTTP(S) Adapter*.

The result XML document for this is structured as follows:

- It contains a base64 encoded string as a MIME body.
- The headers are available as XML structure to easily configure an X4 transfer adapter.
- The generated document is in MIME XML format of type *Part*. Therefore, you can use it to create other multipart documents.

Use case: Web service calls that require multipart documents

Input: Example of a MIME XML document

```
<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: Example of a generated MIME XML document

```
<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

The `PartFromX4Document` operation converts the adapter input as the body of a Mime document (in the XML format of this adapter). A `Content-Type` Header is created with the corresponding values from the X4 document (these values can be influenced with the Document Properties Changer).

Convert X4Document properties to the Content-Type Header

X4Document *mimeType*	X4Document *encoding*	resulting `Content-Type` header
empty	*empty*	*no Header*
empty	`someCharset`	*no Header*
`some/type`	*empty*	`Content-Type: some/type`
`some/type`	`someCharset`	`Content-Type: some/type; charset=someCharset`

This operation is similar to the following adapter sequence:

1. Base64 (encode) - Wrap input as Base64 string
2. ElementWrapper (wrap) - Wrap Base64 string into XML document
3. XSL - Put XML document in the appropriate form (and create the headers)

10.8.3 MIME Multipart Reader

`MIME Multipart Reader`: Reads body parts of a multipart message and outputs one or all body parts as text or as a Base64-encoded XML document with header information.

Properties

`Operation`	Determines which operation the adapter executes **Possible values:** • *`GetDocument:`* Reads a specific part of the multipart message specified in the `index` parameter. • *`Traverse`*: Reads in all parts of the multipart message sequentially in a loop using a `Condition` module. As soon as all parts have been read, the adapter returns the status `0`.

Parameter

`index`	Index (position) of the part to be read in for the `GetDocument` operation. **Possible values:** • Any positive integer (0 = default)
`toXML`	Output all header information of the multipart message as elements and the content of the read part Base64-encoded in an XML document. **Possible values:** • *`yes`*: Output XML document with header information and body part • *`no`*: Output read multipart message as text (default)

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

10.8.3.1 Sample

The adapter expects a multipart message as 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

The operation `GetDocument` reads the following part if the `index` parameter is set to `0` and `toXML` is set to `yes`:

```
This is the body text at index 0
```

10.8.3.3 Traverse

The operation `Traverse` reads all parts of the multipart message in a loop one after the other with the help of a `Condition` module. The following example has the following parameter configuration: `index` = `0` and `toXML` = `yes`.

In the 1st loop pass the first part (header information and body part) is read in as a base64 encoded element:

```
<?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>
```

In the 2nd loop pass the second part (header information and body part) is read in as a Base64 encoded element:

```
<?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`: Sends and receives emails via SMTP (Simple Mails Transfer Protocol) or POP3 (Post Office Protocol v3). The adapter supports multiple file attachments and MIME type definitions for the email body e.g. for sending HTML emails. In addition, the sending and receiving of S/MIME encoded and/or signed email messages is supported.

Former name until version 5.5.4: *POP/SMTP Connector*

Properties

`I/O Mode (input_type)`	Data handling mode **Possible values:** • *`Actions`*: Process data from the preceding process step • *`Directory`*: Process all files of a specified directory as attachment (or files matching a filter within the parameter filter)

`File name (documentname)`	Name of the file attachment to be sent (if `I/O Mode` is set to *`Actions`*).
	
	ⓘ Please note: If you want to parameterize this property dynamically, use its technical name in the corresponding XSL mapping. The technical name may differ from the displayed name.
	
	Example:
	
	`<xsl:processing-instruction name="Dynamic Parameter">`
	`  documentname="`
	`  <xsl:value-of select="."/>`
	`  "`
	`</xsl:processing-instruction>`
	
	Possible values:
	
	• Any file name (e.g. *`Data.xml`*)
	• Multiple file names separated by semicolon (if you hand over multiple input documents to the transfer adapter using `Fork/Join`, they will be processed consecutively according to the branch priorities, e.g. *`File1.xml`*;*`File2.xml`*)

`Delete (delete)`	Delete email message after the successful reception from the POP3 server (if property `Operation` is set to *`Receive`*)
`Operation (action)`	Defines which operation will be executed for the specified file(s). Depending on the value in `I/O Mode`, the function differs. **Possible values:** • *`Send`*: Send email using an SMTP server • *`Receive`*: Receive email from a POP3 server

Parameters

`from`	Email address of the email sender; If required, with specification of a display name **Possible values:** Any valid email address (e.g. *`name@domain.tld (SampleName)`*)
`sender`	Technical sender agent (if different from parameter `from`) which actually sends the message; If required, with specification of a display name **Possible values:** Any valid email address (e.g. *`name@domain.tld (SampleName)`*)
`to`	Email address(es), to which the email shall be sent; If required, with specification of a display name **Possible values:** Any valid email address; in case of multiple addresses separated by semicolon (e.g. *`name@domain.tld;email@host.tld (SampleName)`*)
`cc`	Email address(es) to which the email will be sent as carbon copy; If required, with specification of a display name **Possible values:** Any valid email address; in case of multiple addresses separated by semicolon (e.g. *`name@domain.tld (SampleName);email@host.tld`*)
`bcc`	Email address(es) to which the email will be sent as blind carbon copy; If required, with specification of a display name **Possible values:** Any valid email address; in case of multiple addresses separated by semicolon (e.g. *`name@domain.tld;email@host.tld (SampleName)`*)
`replyTo`	Email address to which the reply shall be sent **Possible values:** Any valid email address (e.g. *`name@domain.tld`*)
`subject`	Subject of the email to be sent
`body`	Message body that represents the email's content (if the input document of the adapter is an HTML document, it will be used as HTML message body) **Possible values:** Any string

`mimeTypeBody`	MIME type of the email body (allows to display HTML emails or multiple attachments) **Possible values:** Valid MIME type (e.g. *`multipart/mixed`*)
`charset`	Encoding of the email body **Possible values:** Valid encoding/charset (e.g. *`ISO-8859-15`*)
`smtp`	IP address or host name of the SMTP server to send emails **Possible values:** Any valid string (e.g. *`smtp.domain.tld`*)
`smtpPort`	Port of the SMTP server to send emails **Possible values:** Any positive integer number, e.g. *`250`*
`smtpSecurityProtocol`	SMTP security protocol to be used **Possible values:** • `NONE`: Do not use any security protocol • `TLS`: Use TLS as security protocol • `SSL`: Use SSL as security protocol
`smtpAllowAllHosts`	Allow all hosts for SMTP ⚠ Please note that hosts without a valid certificate will also be allowed. ℹ This parameter is only valid, if SSL is used as security protocol.
`pop3`	IP address or host name of the POP3 server to receive emails **Possible values:** Any valid string (e.g. *`pop3.domain.tld`*)
`pop3Port`	Port of the POP3 server to receive emails **Possible values:** Any positive integer number
`pop3SecurityProtocol`	POP3 security protocol to be used **Possible values:** • `NONE`: Do not use any security protocol • `TLS`: Use TLS as security protocol • `SSL`: Use SSL as security protocol

`pop3AllowAllHosts`	Allow all hosts for POP3 ⚠ Please note that hosts without a valid certificate will also be allowed. ℹ This parameter is only valid, if SSL is used as security protocol.
`login`	User name for the POP3 server (when receiving email) or for the SMTP server (when sending an email) **Possible values:** Any valid string (e.g. *`name@domain.tld`*)
`password`	Password for the POP3 server (when receiving email) or for the SMTP server (when sending an email)
`oAuth2`	Defines if the OAuth 2.0 protocol is used for log-in. **Possible values:** • *true*: `oAuth2` is used for log-in • *false*: `oAuth2` is not used for log-in ⚠ If the OAuth 2.0 protocol is used, • the SSL encryption must be enabled for the connection to the POP server by setting the `ssl` parameter. • the access token must be entered as the `password` parameter.
`oAuth2_twoLineAuth`	Activates the POP3 protocol for Microsoft 365. ⚠ Needs to be activated in addition to `oAuth2` to use POP3 protocol with Microsoft 365. **Possible values:** • *true*: `oAuth2_twoLineAuth` is used in addition to `oAuth2` • *false*: `oAuth2_twoLineAuth` is not used in addition to `oAuth2`

`file_limit`	Limit the number of attachments to be sent (if `I/O Mode` is set to *`Directory`* and thus multiple files of a directory should be attached) **Possible values:** • Any positive integer number • *0*: Do not attach files (Default)
`dir`	Local directory on the *X4 Server* e.g. in the X4 Repository that contains files to be sent as email attachments (if `I/O Mode` is set to *`Directory`*) **Possible values:** Any string (e.g. *`%USER_COLLECTION%\Folder`*)
`filter`	Filter files to be send as attachments by their name using the placeholders `*` and `?` **Possible values:** • Any string, e.g. *`*.dat`* applies only for files with the extension `.dat` • *: Use all files (do not filter) (Default)
`EML`	Send a message in EML format (this requires an EML message as adapter input) or receive a message in EML format **Possible values:** • *`true`*: Send in EML format • *`false`*: Use standard email format (Default)
`base64`	Encode attachments (multipart content) always as base64 (if the property `Operation` is set to *`Receive`*) **Possible values:** • *`true`*: Encode all multipart contents as Base64 • *`false`*: Do not encode as base64 (Default)

`encrypt`	Encrypt the message with S/MIME (*Multipart/Encrypted*) **Possible values:** • *`true`*: Enable S/MIME encryption (requires to specify a key/certificate file within the parameter `Pkcs12` and a passphrase in parameter `passphrase`) • *`false`*: No encryption (Default)
`sign`	Sign the message with S/MIME (*Multipart/Signed*) **Possible values:** • *`true`*: Add S/MIME signature (requires to specify a key/certificate file within the parameter `Pkcs12` and a passphrase in parameter `passphrase`) • *`false`*: No signature (Default)
`Pkcs12`	Path to a valid key/certificate in the *PKCS#12* format **Possible values:** Any valid path in the local file system of the X4 Server (e.g. *`%USER_COLLECTION\Project\Folder\Key.p12`*)
`passphrase`	Password/passphrase to the key/certificate file in *PKCS#12* format that has been specified within the parameter `Pkcs12` **Possible values:** Any valid string
`propertiesType`	Date format for a properties file which can be used to define further *JavaMail* properties **Possible values:** • *`text`*: Properties file in text format • *`xml`*: Properties file in XML format, see http://www.ibm.com/developerworks/library/j-tiger02254/index.html

`propertiesUrl`	**Deprecated Parameter** As of version 6.2.0 of the X4 BPMS, the security protocol to be used can be set via the parameters `smtpSecurityProtocol` and `pop3SecurityProtocol`. If using the now deprecated parameter `propertiesURL`, a warning will be displayed in the server log. URL to a properties file which can be used to define further *JavaMail* properties, see https://javamail.java.net/nonav/docs/api/com/sun/mail/smtp/package-summary.html. The linked file must be available in the format defined within parameter `propertiesType`. If the `propertiesURL` is set, the parameters defined within the properties file are used. Otherwise the adapter parameters will be used. **Possible values:** Any valid URL, e.g. `xstore://Project/Folder/File.properties`

Which e-mail addresses are formally valid can be found in the guidelines *RFC 5321* and *RFC 5322*.

10.9 ERP

- Abas ERP Connector
- SAP Connector

10.9.1 Abas ERP Connector

`Abas ERP Connector`: Connects to a Abas ERP system via its *jedp* interface and executes different operations.

Properties

`Operation`	Defines the operation executed by the connector **Possible values:** • `Query`: query data records and table contents • `Info`: query metadata from abas • `Write`: add or change data records and table contents

Parameter

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.abas.AbasAdapter*: Main class (Default)
appName	Application name **Possible values:** Any string
edpPort	EDP port **Possible values:** Any integer, e.g. `6550`
mandant	Client name **Possible values:** Any string, e.g. `systemtest`
host	Server IP address or Host name **Possible values:** Any valid string without protocol specification, e.g. `example.com`
password	Password for the authentication **Possible values:** Any string
edpVersion	EDP version **Possible values:** Any integer, e.g. `3.35` (default)
wait	Lock behavior of the session **Possible values:** • `WAITINFINITE`: wait until lock is released • `WAIT`: wait until lock is released or timeout is reached • `NOWAIT`: return error instantly and without waiting (default)
timeout	Timeout value for the connection setup in seconds **Possible values:** Any integer, e.g. `30` (default)

Status Values

`1 (OK)`	The execution was successful

-1 (ERROR)	An error occurred during execution, e.g. due to an invalid XML input
0 (TIMEOUT)	A timeout occurred while querying the Abas server

Input

This adapter expects the following XML input structures for the different operations:

Operation: Query

```
<Abas>
    <!-- Execute query -->
    <Query resulttag="...">
        <DatabaseGroup>00:01</DatabaseGroup>
        <Fields>
            <_addr/>
            <_str/>
            <_number/>
        </Fields>
        <Selection>number=20001!20002</Selection>
    </Query>
    <Query resulttag="...">
        <DatabaseGroup>01:01</DatabaseGroup>
        <Fields>
            <_schlag/>
            <_contractnumber/>
            <_client/>
        </Fields>
        <Selection>client=20001!20002</Selection>
    </Query>
</Abas>
```

Operation: Info

```
<Abas>
    <!-- Execute action -->
    <Info resulttag="...">
        <Number>1000</Number>
        <HeaderFields>
            <_bstart>1</_bstart>
        </HeaderFields>
        <TableFields>
            <_tysearch/>
            <_tyname/>
        </TableFields>
    </Info>
    <Info resulttag="...">
        <Number>1000</Number>
        <HeaderFields>
            <_bstart>1</_bstart>
        </HeaderFields>
        <TableFields>
            <_tysearch/>
        </TableFields>
    </Info>
</Abas>
```

Operation: Write

```
<Abas>
    <!-- Change existing dataset -->
    <Update resulttag="...">
        <Database>00</Database>
        <Group>01</Group>
        <Number>20001</Number>
        <HeaderFields>
            <_str>Test street</_str>
        </HeaderFields>
    </Update>
    <!-- New dataset -->
    <Insert resulttag="...">
        <Database>00</Database>
        <Group>01</Group>
        <HeaderFields>
            <_number>200171</_number>
            <_search>Test search</_search>
            <_addr2>Test addr</_addr2>
            <_str2>Test str</_str2>
            <_zipcode2>12345</_zipcode2>
            <_nort2>Testhausen</_nort2>
            <_zbed>100</_zbed>
        </HeaderFields>
    </Insert>
    <!-- Change line in existing dataset -->
    <Update resulttag="...">
        <Database>71</Database>
        <Group>02</Group>
        <Number>4</Number>
        <Table>
            <Row nr="2">
                <TableFields>
                    <_ytg02description>blue</_ytg02description>
                </TableFields>
            </Row>
        </Table>
    </Update>
    <!-- New line in existing dataset -->
    <Update resulttag="Testupdate3">
        <Database>71</Database>
        <Group>02</Group>
        <Nummer>4</Nummer>
        <Table>
            <Row nr="2" new="true">
                <TableFields>
                    <_ytg02description>yellow</_ytg02description>
                </TableFields>
            </Row>
        </Table>
    </Update>
</Abas>
```

Output

This adapter outputs the following XML result documents for the different operations:

Operation: Query

```
<Result>
    <Address>
        <_addr>Max Mustermann</_addr>
        <_str>Musterallee 2</_str>
        <_number>20001</_number>
    </Address>
    <Address>
        <_adr>Manfred Mustermann</_adr>
        <_str>Testweg 1</_str>
        <_number>20002</_number>
    </Address>
    <Contract>
        <_schlag>RECHNUNG.2017.10</_schlag>
        <_ycontractnumber>01-000001-001</_ycontractnumber>
        <_client>123</_client>
    </Contract>
    <!-- More nodes -->
</Result>
```

Operation: Info

```
<Result>
    <Test>
        <_tysearch>XY123</_tysearch>
        <_tyname>Test Abas-Server</_tyname>
    </Test>
    <Test2>
        <_tysearch>XY123</_tysearch>
    </Test2>
</Result>
```

Operation Write

```
<Result>
    <Testupdate1>updated</Testupdate1>
    <Testinsert1>inserted</Testinsert1>
    <Testupdate2>updated</Testupdate2>
    <Testupdate3>updated</Testupdate3>
</Result>
```

Dependencies

The connector uses library *jedp* by abas, `jedp-3.3.5.jar`

The currently supported version is 3.3.5.
The library is not delivered for licensing reasons. Copy the library file to X4 Server installation folder `wildfly\modules\system\layers\base\de\softproject\x4\extensions\main\` together with a file `module.xml` with the following contents:

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 structures

How IDocs in Raw format and XML IDocs are structured

IDoc in Raw format

An IDoc Raw document is an XML document that contains metadata about the message, structure information, and SAP IDoc in raw format as text. The XML document is structured as follows:

```
<X4SAP TID="<!--Transaction ID-->">
    <X4IDoc type="<!--IDoc type-->" description="">
        <X4Mandatory>
            <X4RecipientPort><!--Recipient port--></X4RecipientPort>
            <X4RecipientPartnerType><!--Partner type abbreviation--></
X4RecipientPartnerType>
            <X4RecipientPartnerNumber><!--Partner number--></
X4RecipientPartnerNumber>
            <X4RecipientPartnerFunction />
            <X4RecipientAdress />
            <X4RecipientLogicalAdress />
            <X4SenderPort><!--Sender port--></X4SenderPort>
            <X4SenderPartnerType><!--Partner type abbreviation--></
X4SenderPartnerType>
            <X4SenderPartnerNumber><!--Partner number--></X4SenderPartnerNumber>
            <X4SenderPartnerFunction />
            <X4SenderAddress />
            <X4SenderLogicalAddress />
            <MessageType> <MSGTYPE><!--Message type--></MSGTYPE> </MessageType>
        </X4Mandatory>
        <X4Structure>
            <Tablename SEGMENT="1"> <!--Segment 1--> </Tablename>
        </X4Structure>
        <RAW> <!--Raw data of the IDoc message (text)--> </RAW>
    </X4IDoc>
</X4SAP>
```

IDoc in XML format

An IDoc XML document contains one IDoc message that has been completely converted to XML elements. In addition, the XML document contains metadata about the message and structure information. Within the element `<IDOC>` all tables are successively deposited.

An IDoc in XML format is structured as follows:

```
<X4SAP TID="<!--Transaction ID-->">
  <X4IDoc type="<!--IDoc type-->" description="">
    <X4Mandatory>
      <X4RecipientPort><!--Recipient port--></X4RecipientPort>
      <X4RecipientPartnerType><!--Partner type abbreviation--></
X4RecipientPartnerType>
      <X4RecipientPartnerNumber><!--Partner number--></X4RecipientPartnerNumber>
      <X4RecipientPartnerFunction />
      <X4RecipientAdress />
      <X4RecipientLogicalAdress />
      <X4SenderPort><!--Sender port--></X4SenderPort>
      <X4SenderPartnerType><!--Partner type abbreviation--></X4SenderPartnerType>
      <X4SenderPartnerNumber><!--Partner number--></X4SenderPartnerNumber>
      <X4SenderPartnerFunction />
      <X4SenderAddress />
      <X4SenderLogicalAddress />
      <MessageType>
        <MSGTYPE><!--Message type--></MSGTYPE>
      </MessageType>
    </X4Mandatory>
    <X4Structure>
      <Tablename SEGMENT="1">
        <!--Segment 1-->
      </Tablename >
    </X4Structure>
    <RAW>
       <!--Raw data of the IDoc message (text)-->
    </RAW>
  </X4IDoc>
</X4SAP>
```

10.9.2.2 SAP Connector (JCo 2)

`SAP Connector (JCo 2)`: Enables the transmission of data from the X4 ESB to an *SAP R/3* system. It retrieves any Remote Function Call (RFC) or any BAPI function module in an SAP system. In addition, IDocs can be sent to *SAP R/3*.

Former designation up to version 4.5.1: *SAP (JCo 2) Request Adapter*

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Execute`*: Execute adapter

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.x4.sapadapter.SAPAdapter`*: Main class (default)
x4.connectionPoolEnabled	Use Connection Pooling
x4.connectionPoolUID	ID of the connection pool. The SAP adapter can use an already existing connection pool if the same connection parameters and the same connection pool ID have been set.
x4.connectionPoolMax	Maximum number of defined connections in the connection pool
x4.transactionEnabled	Enable or disable transactional processing.
x4.transactionID	Transaction ID; all operations of the SAP adapter within a process are executed as one transaction and have the same transaction ID. The SAP adapter can use a transaction that has already been started if they have the same transaction parameters and the same transaction ID.
jco.client.client	SAP-specific parameter: SAP client
jco.client.user	SAP-specific parameter: User name
jco.client.passwd	SAP-specific parameter: Password
jco.client.lang	SAP-specific parameter: Logon language
jco.client.sysnr	SAP-specific parameter: SAP system number
jco.client.ashost	SAP-specific parameter: SAP application server
jco.client.mshost	SAP-specific parameter: SAP message server
jco.client.gwhost	SAP-specific parameter: Gateway Host
jco.client.gwserv	SAP-specific parameter: Gateway Service
jco.client.r3name	SAP-specific parameter: R/3 name
jco.client.group	SAP-specific parameter: SAP application server group

jco.client.tpname	SAP-specific parameter: Programme ID of an external server application
jco.client.tphost	SAP-specific parameter: Host of the external server application
jco.client.type	SAP-specific parameter: Type of remote host **Possible values:** • *2*: R/2 • *3*: R/3 • *E*: External
jco.client.trace	SAP-specific parameter: RFC trace **Possible values:** • *0*: Deactivate RFC trace • *1*: Activate RFC trace
jco.client.codepage	SAP-specific parameter: Initial code page in SAP notation
jco.client.abap_debug	SAP-specific parameter: ABAP debugging **Possible values:** • *0*: Deactivate ABAP debugging • *1*: Activate ABAP debugging
jco.client.use_sapgui	SAP-specific parameter: Name of an SAP graphical user interface to be used.
jco.client.getsso2	SAP-specific parameter: SSO ticket received after login **Possible values:** • *0*: No SSO ticket received • *1*: SSO ticket received after registration
jco.client.mysapsso2	SAP-specific parameter: Name of the SAP cookie version 2 to be used as a logon ticket.
jco.client.x509cert	SAP-specific parameter: Name of the X509 certificate to be used as the logon ticket.

jco.client.lcheck	SAP-specific parameter: Logon check on opening **Possible values:** • *0*: Deactivate login verification • *1*: Enable login verification when opening
jco.client.grt_data	SAP-specific parameter: Additional data for the SAP graphical user interface
jco.client.use_guihost	SAP-specific parameter: Host to which the remote SAP graphical user interface is to be forwarded.
jco.client.use_guiserv	SAP-specific parameter: Service to which the remote SAP graphical user interface is to be forwarded.
jco.client.use_guiprogid	SAP-specific parameter: Programme ID of the server that starts the remote SAP graphical user interface.
jco.client.snc_mode	SAP-specific parameter: *Secure network connection* (SNC)-Mode **Possible values:** • *0*: Deactivate SNC mode • *1*: Activate SNC mode
jco.client.snc_partnername	SAP-specific parameter: SNC partner, e.g. *p:CN=R3, O=XYZ-INC, C=EN*
jco.client.snc_qop	SAP-specific parameter: SNC security level **Possible values:** *1...9*: SNC security level
jco.client.snc_myname	SAP-specific parameter: SNC name (overwrites the default SNC partner)
jco.client.snc_lib	SAP-specific parameter: Path to a library that provides the SNC service
jco.client.dest	SAP-specific parameter: R/2 destination
jco.client.saplogon_id	SAP-specific parameter

jco.client.extiddata	SAP-specific parameter: External authentication data (PAS)
jco.client.extidtype	SAP-specific parameter: Type of external authentication (PAS)

A complete description of all (connection) parameters can be found in the *SAP JCo 2* documentation.

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Call RFC/BAPI

Input

The adapter expects an input XML document according to the following pattern, which calls the corresponding function module with input data and the affected tables, if applicable:

```
<X4SAP>
 <bapicall name="BAPI-Name" id="IDoc-ID">
   <input>
      <INPUT>
         <!--Parameter und Parameterwerte -->
      </INPUT>
   </input>
   <table>
      <!--Tabellen-->
   </table>
 </bapicall>
</X4SAP>
```

Example

To call `BAPI IDOCTYPE_READ_COMPLETE` in SAP, this adapter expects the following input XML document:

```
<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>
```

Send IDoc

Input

To send an IDoc to SAP, use an XML document like the following as input to the adapter:

```
<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)

? Unknown Attachment `SAP Connector (JCo 3)`: Enables the transmission of data from the X4 ESB to an *SAP R/3* system. It generates any Remote Function Call (RFC) or calls any BAPI function module in an SAP system via *JCo 3*. In addition, IDocs can be sent to an SAP host.

Former designation up to version 4.5.1: *SAP (JCo 3) Request Adapter*

Properties

Operation	Determines which operation the adapter executes **Possible values:** `Execute`: Execute adapter

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *en.softproject.integration.adapter.sapjco3.SAPAdapter*: Main class (default)
PropertyFile	URL to the configuration file for the adapter **Possible values:** Any URL valid within the Java VM (e.g. `xstore://Project/Folder/Configuration.properties`).
ReplaceEscapedControlChars	Replace masked control characters (e.g.: `&#31;`) in the XML strings returned by the JCo library with the replacement character ( ) **Possible values:** • `true`: Replace masked control characters • `false`: Do not replace masked control characters

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Call RFC/BAPI

Input

The adapter expects an input XML document according to the following pattern, which calls the corresponding function module with input data and the affected tables, if applicable:

```
<X4SAP>
 <bapicall name="BAPI-Name" id="IDoc-ID">
   <input>
      <INPUT>
         <!--Parameter und Parameterwerte-->
      </INPUT>
   </input>
   <table>
      (Tabellen)
   </table>
 </bapicall>
</X4SAP>
```

Example

To call BAPI `IDOCTYPE_READ_COMPLETE` in SAP, this adapter expects the following input XML document:

```
<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>
```

Send IDoc

To send an IDoc to SAP, use an XML document like the following as input to the adapter:

```
<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)`: Creates IDoc and BAPI structure descriptions, lists BAPIs or IDocs, creates empty IDoc and BAPI structures, and outputs an XML document according to the used operation.

Former name until version 4.5.1: *SAP (JCO 2) Explorer Adapter*

> **Please note:**
> To use the *SAP Explorer (JCo 2)*, rights for the remote execution of the following RFC components is required:
>
> - `RPY_BOR_TREE_INIT`
> - `IDOCTYPES_LIST_WITH_MESSAGES`
> - `IDOCTYPE_READ_COMPLETE`
> - `RPY_OBJECTTYPE_READ`
> - `RFC_FUNCTION_SEARCH`

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *`Get BapiTree`*: Output a list with all available BAPIs • *`Get IDocList`*: Output a list with all available IDocs • *`Read Bapi`*: Output a BAPI structure description • *`Read IDoc`*: Output an IDoc structure description • *`Create Empty Bapi`*: Create empty BAPI structure in order to use it within X4 BPMS • *`Create Empty Idoc`*: Create empty IDoc structure in order to use it within X4 BPMS • *`SearchForBapis`*: Output list of all BAPIs that match the filter expression • *`Nothing`*: Output the input unmodified

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.sapadapter.SAPBapiIdocNotationExplorer*: Main class (Default)

PropertyFile	URL to the adapter's configuration file (overrides the adapter configuration) **Possible values:** Any valid URL within the Java VM (e. g. *xstore://Project/Folder/Configuration.properties*)
Client	SAP client
User	User name
Password	Password
Language	Login language
SystemNumber	SAP system number
RouterString	Router string
SystemType	Remote host type **Possible values:** • *2*: R/2 • *3*: R/3 • *E*: External

Input

This function adapter expects an adapter-specific input XML structure depending on the selected operation.

Operation	
Get BapiTree	For the operation `Get BapiTree` the adapter doesn't require any input XML document.
Get IDocList	For the operation `Get IDocList` the adapter doesn't require any input XML document
Read Bapi	For the operation `Read Bapi` the adapter expects the following input XML document: `<Object><!--BAPI-ID--></Object>`

Operation	
Read IDoc	For the operation `Read IDoc` the adapter expects the following input XML document: `<ReadIDoc><!--IDoc-ID--></ReadIDoc>`
Create Empty Bapi	For the operation `Create Empty Bapi` the adapter expects the following input XML document: `<Bapiname><!--BAPI name--></Bapiname>`
Create Empty Idoc	For the operation `Create Empty Idoc` the adapter expects the following input XML document: `<IDoc><!--IDoc name--></IDoc>`
SearchForBapis	For the operation `SearchForBapis` the adapter expects the following input XML document: `<Search functionname="BAPI name/Function name" />`
Nothing	For the operation `Nothing` the adapter doesn't require any input XML document

10.9.2.5 SAP Explorer (JCo 3)

`SAP Explorer (JCo3)`: Generates structure descriptions of IDocs and BAPIs, lists BAPIs or IDocs, or generates empty IDoc and BAPI structures. Depending on the operation, it outputs a corresponding XML document.

Former name until version 4.5.1: *X4 SAP (JCo 3) Explorer Adapter*

Please note:
To use the *SAP Explorer (JCo 3)*, rights for the remote execution of the following RFC components is required:

- `RPY_BOR_TREE_INIT`
- `IDOCTYPES_LIST_WITH_MESSAGES`
- `IDOCTYPE_READ_COMPLETE`
- `RPY_OBJECTTYPE_READ`
- `RFC_FUNCTION_SEARCH`

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *`GetBapiTree`*: Output a list with all available BAPIs • *`GetIDocList`*: Output a list with all available IDocs • *`ReadBapi`*: Output a BAPI structure description • *`ReadIDoc`*: Output an IDoc structure description • *`CreateEmptyBapi`*: Create empty BAPI structure in order to use it within X4 BPMS • *`CreateEmptyIdoc`*: Create empty IDoc structure in order to use it within X4 BPMS • *`SearchForBapis`*: Output list of all BAPIs that match the filter expression

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.sapjco3.SAPBapiIdocNotationExplorer*: Main class (Default)
PropertyFile	URL to the adapter's configuration file **Possible values:** Any valid URL within the Java VM (e. g. *xstore://Project/Folder/Configuration.properties*)
Client	SAP client (`jco.client.client` within the properties file)
User	User name (`jco.client.user` within the properties filei)
Password	Password (`jco.client.passwd` within the properties file)
Language	Login language (`jco.client.lang` within the properties file)
SystemNumber	SAP system number (`jco.client.sysnr` within the properties file)

RouterString	Router string (`jco.client.ashost` within the properties file)
SystemType	Remote host type (`jco.client.type` within the properties file) **Possible values:** • *2* : R/2 • *3* : R/3 • *E* : External
ReplaceEscapedControlChars	Replace masked control characters (e.g.: *&#31;*) within the XML strings returned by the JCo library by the replacement character ( ) **Possible values:** • `true`: Replace masked control characters • `false`: Do not replace masked control characters

Input

This function adapter expects an adapter-specific input XML structure depending on the selected operation.

Operation	
Get BapiTree	For the operation `Get BapiTree` the adapter doesn't require any input XML document.
Get IDocList	For the operation `Get IDocList` the adapter doesn't require any input XML document
Read Bapi	For the operation `Read Bapi` the adapter expects the following input XML document: `<Object><!--BAPI-ID--></Object>`
Read IDoc	For the operation `Read IDoc` the adapter expects the following input XML document: `<ReadIDoc><!--IDoc-ID--></ReadIDoc>`

Operation	
Create Empty Bapi	For the operation `Create Empty Bapi` the adapter expects the following input XML document: `<Bapiname><!--BAPI name--></Bapiname>`
Create Empty Idoc	For the operation `Create Empty Idoc` the adapter expects the following input XML document: `<IDoc><!--IDoc name--></IDoc>`
SearchForBapis	For the operation `SearchForBapis` the adapter expects the following input XML document: `<Search functionname="BAPI name/Function name" />`

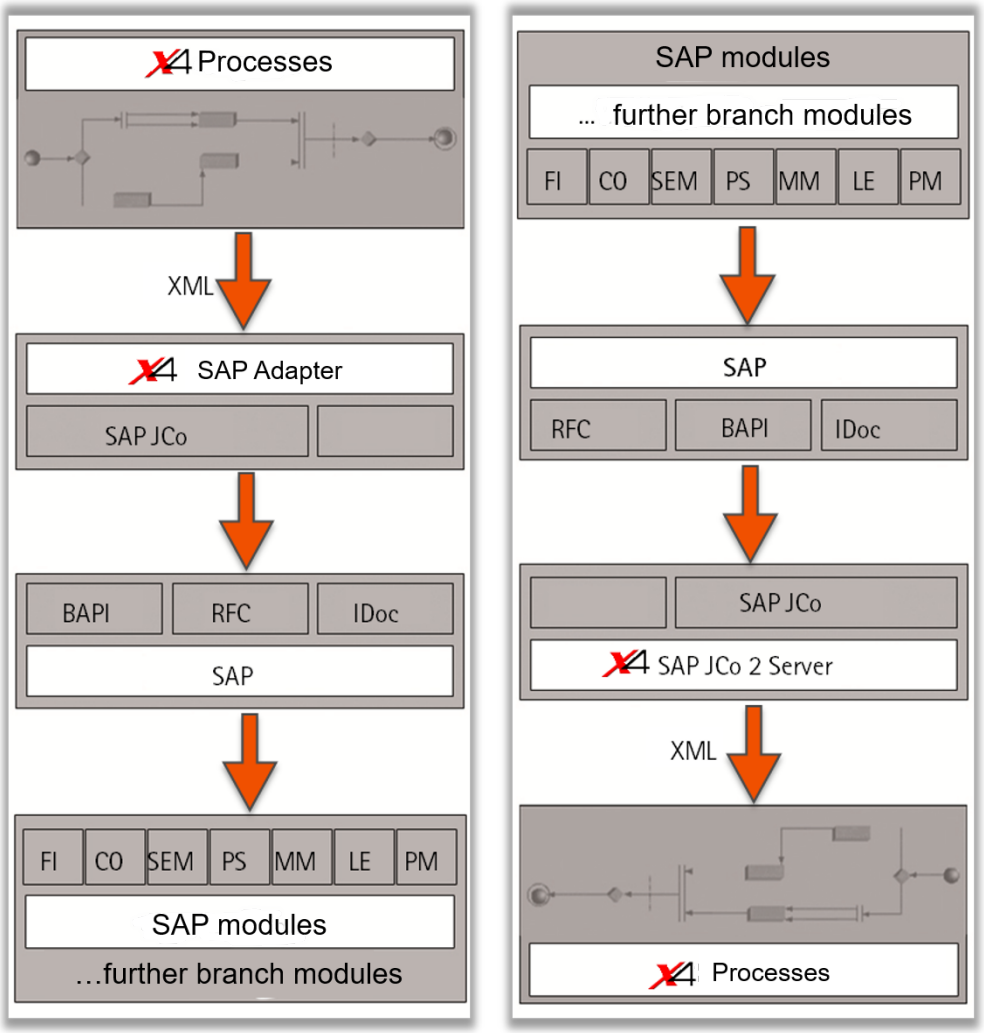
10.9.2.6 SAP Listener (JCo 2)

The *SAP Listener (JCo 2)* allows the processing of incoming communication requests via BAPI/RFC and of IDocs of an SAP system. If necessary, it automatically re-establishes broken connections. Former name until version 4.5.1: *X4 SAP JCo 2 Listener*

Functions

The *X4 Server* allows the bidirectional integration of *SAP R/3* systems using the *SAP Java Connector* in version 2 (*JCo 2*). The following components are relevant:

- *SAP JCo 2 Server*: Allows the processing of incoming communication requests via BAPI/RFC and of IDocs of an SAP system. If necessary, it automatically re-establishes broken connections.
- *SAP Connector (JCo 2)*: In *X4* processes, this function adapter can invoke any Remote Function Call (RFC) or any BAPI function module in an SAP system. Moreover, IDocs can be sent to *SAP R/3*.
- *SAP Explorer (JCo 2)*: This adapter generates structure descriptions of IDocs and BAPIs, lists BAPIs or IDocs, or generates empty IDoc and BAPI structures. Depending on the operation, it outputs a corresponding XML document.



SAP Java Connector

Using the *JCo 2 Server*, multiple SAP systems can be connected via a respective separate *JCo 3 Server* instance. To use the *JCo 2 Server* a corresponding configuration is required:

- *X4* configuration within the `X4config.xml`; see SAP JCo 2 server configuration
- Definition of connection properties; see Properties for the JCo 2 server
- Assignment of the function block calls to Technical Process; see Function block/method definitions (X4 SAP JCo 2)

Identification of JCo 2 Servers via ProgID

Each *JCo 2 Server* instance or any system connected to SAP communicates via a unique *ProgID*, which is configured within the server properties file, by default `jco_server_settings.prop` and is introduced to the SAP system at the start of the server instance.

Several *ProgIDs* can be used simultaneously for each SAP system. Using the *ProgID*, IDocs and RFC/BAPIs are "addressed" to the respective target system.

Installing X4 SAP JCo 2

Which steps are necessary to install *X4 SAP JCo 2*

> As an example, an installation for the JBoss application server based on a 32 bit Windows system will be described.

Prerequisites

To install an *SAP JCo 2 Server* within the *X4 Server* SAP libraries are needed, which are operating system and architecture-dependent. These libraries are also needed by the *SAP Connector (JCo 2)* and the *SAP Explorer (JCo 2)*.

As a registered SAP partner, you can download the necessary libraries and further documentation via *SAPNet* under http://service.sap.com/connectors.

> Make sure you select mutually compatible version numbers of the libraries (these are also operating system and architecture-dependent).
> Make sure that your operating system is compatible with the used JDK version and the version of your SAP system. The following matrix shows the relationship of these three variables.

		OS		SAP	
		32 bit	64 bit	32 bit	64 bit
JDK	32 bit	x	x	x	-
	64 bit	-	x	-	x

Installing the libraries

1. Copy the native libraries (depending on the system) to the respective folder of the application server for native binary dependencies or to the `bin` directory of Java.
   Example: Under Windows 32 bit: Copy `librfc32.dll`, `msvcr71.dll`, `msvcp71.dll` and `sapjcorfc.dll` to `C:\Windows\System32` or to `C:\X4\Server\jdk1.7\bin`.
2. Copy the following Java libraries to the classpath of the application server:
   - `sapidoc.jar`
   - `sapjco.jar`
   - `sapidocjco.jar`

   Example JBoss 4.2.3.GA: Copy these libraries to `C:\X4\JBoss4.2.3.GA\server\default\lib`.
   Example JBoss AS 7.1.1: Copy these libraries under `C:\X4\Server\jboss-as-7.1.1.Final\standalone\deployments` to the `lib` directory of `X4.ear`.
3. Restart the *X4 Server*.

Testing the installation

- Test the connection using the *SAP Explorer (JCo 2)* (analogous to the *X4 Experience* example `X4Experience/AdapterSuites/ X4SAPSuite// Processes/ CreateEmptyBapi.wrf`)
- Make sure that the configuration of the *SAP JCo 2 Server* is complete
- Set the adapter parameter `SystemType` to *3* (for a connection to *SAP R/3*)
- Make sure that the adapter operation `Create Empty Bapi` is set
- Execute the process with the input XML `<Bapiname>BAPI's name</Bapiname>` and check the result XML

Configuring the SAP Server (JCo 2)

Learn which configuration has to be made to start the *JCo 2 Server*, to make it accessible for external access and to the assign the appropriate handler processes within the X4 Server to the incoming RFC/BAPIs and IDocs.

SAP JCo 2 server configuration

Within the element `<jcoServer>` of the `X4config.xml` you can configure the *SAP Java Connector* settings for the *X4 Server*.

> Beginning with the *X4 Server* revision *27282* the element `<jcoServer>` has an altered structure: IDoc handlers and Function handlers have been outsourced in the *JCo 2 Server Properties*, see Properties for the JCo 2 server. Moreover, is now possible to specify multiple properties files and function block/method definitions within the `X4config.xml` to allow connections to multiple SAP systems.
> If multiple properties files and function block/method definitions are specified, their assignment takes place based on the order, i. e. the first *JCo Server* properties file in `<propertiesURL>` is assigned to the first function blockmethod definition in `<methodsURL>`, the second *JCo Server* properties file to the second function block/method definition etc.

Element <jcoServer>

`<propertiesURL>`	URL to a properties file containing the *JCo 2 Server* settings; Beginning with *X4 Server* Version 4.2, you can specify multiple URLs to different properties files separated by semicolons **Possible values**: Any valid URL (e. g. *`file:///C:/X4/X4DB/0/jco_server_settings.prop`*)

`<methodsURL>`	URL to an XML document containing the assignment of function modules/methods and the corresponding processing process in the *X4 Server*; Beginning with *X4 Server* Version 4.2, you can specify multiple URLs to various configuration documents separated by commas **Possible values**: Any valig URL (e. g. *`file:///C:/X4/X4DB/0/jco_method_def.xml`*)

Element <sapadapter>
Within the element `<sapadapter>`, which is a following sibling of the element `<jcoServer>` in the `X4config.xml`, you can configure the default connection settings for the *SAP Connector (JCo 2)*. These will be used as fallback configuration.

`<propertiesURL>`	URL to a properties file containing connection settings that are used when the *SAP Connector (JCo 2)* does not find any settings.

Properties for the JCo 2 server

Within a configuration document (by default: `<X4>/X4DB/0/jco_server_settings.prop`) the number of *JCo Server* instances, function handler processes and IDoc handler processes are defined. For each *JCo Server* instance a JCo Server configuration file must be referenced in the central configuration file `<X4>/X4config.xml` . The properties file is divided into the following blocks.
General configuration
Regarding the allowed server instances and the handling with connections and Connection Pooling, the JCo 2 Server can be configured with the following properties, see Connection pooling (X4 SAP JCo 2).

```
# Number of server instances
server_instances_count = Number of JCo Server instancen
usePool                = Boolean value
poolSize               = Permitted number of connections
```

Explanation:

- `server_instances_count`: Number of *JCo Server* instances
- `usePool`: Activate (*true*) oder deactivate (*false*) Connection Pooling, see Connection pooling (X4 SAP JCo 2)
- `poolSize`: Number of permitted client connections in the pool; Recommendation: Choose a value corresponding at least to the sum of the *JCo 2 Server* instancen (identified by a *ProgID*).

Function handler configuration
In the following properties the function handlers for RFC/BAPI calls per SAP host are defined. The configuration of the default function handler processes is optional and is only used as a fallback, if no appropriate handler processes were assigned in the function block/method definitions, see Function block/method definitions (X4 SAP JCo 2).

However, commit and rollback handlers can only be agreed using these properties.

```
# per jcoserver instance settings
# this should always be set in the method definitions
jcoserver.function.handler.userId   = User folder within the X4DB (optional)
jcoserver.function.handler.process  = Default Function Handler process (optional)

jcoserver.function.commit.userId    = User repository for the commit process
jcoserver.function.commit.process   = Path to the X4 process for commit
jcoserver.function.rollback.userId  = User repository for rollback processes
jcoserver.function.rollback.process = Path to the X4 process for rollback
```

Explanation:

- `jcoserver.function.handler.userId`: User repository within the `X4DB` (e. g. *1*), in which the function call handler process is located
- `jcoserver.function.handler.process`: Optional path specification to an *X4* process serving as default handler for function calls
- `jcoserver.function.commit.userId`: User repository within the `X4DB` (e. g. *1*), in which the *Commit* handler process for function calls is located
- `jcoserver.function.commit.process`: Path specification to an *X4* process serving as *Commit* handler for function calls
- `jcoserver.function.rollback.userId`: User repository within the `X4DB` (e. g. *1*), in which the *Rollback* handler process for function calls is located
- `jcoserver.function.rollback.process`: Path specification to an *X4* process serving as *Rollback* handler for function calls

IDoc processing
In the following properties the handlers for incoming iDocs are defined. The configuration of the default iDoc handler processes is optional and is only used as a fallback, if no appropriate handler processes were assigned in the function block/method definitions, see Function block/method definitions (X4 SAP JCo 2).

For each *JCo 2 Server* instance (identified by a respective *ProgID*) only a commit and rollback handler can be agreed using these properties.

ⓘ If no handler configuration was found, the corresponding Commit respectively Rollback will be skipped.

```
# this should always be set in the method definitions
jcoserver.idoc.handler.userId   = User folder within the X4DB (optional)
jcoserver.idoc.handler.process  = Default iDoc handler process (optional)

# commit and rollback handlers for IDocs
jcoserver.idoc.commit.userId    = User repository for the Commit process
jcoserver.idoc.commit.process   = Path to the X4 process for the Commit
jcoserver.idoc.rollback.userId  = User repository for the Rollback process
jcoserver.idoc.rollback.process = Path to the X4 process for the Rollback
```

Explanation:

- `jcoserver.idoc.handler.userId`: Optional specification of the user repository within the `X4DB` (e. g. *1*), in which the iDoc handler process is located
- `jcoserver.idoc.handler.process`: Optional path specification to an *X4* process serving as default handler for iDocs (Fallback)
- `jcoserver.idoc.commit.userId`: User repository within the `X4DB` (e. g. *1*), in which the *Commit* handler process for iDocs is located
- `jcoserver.idoc.commit.process`: Path specification to an *X4* process serving as *Commit* handler for iDocs
- `jcoserver.idoc.rollback.userId`: User repository within the `X4DB` (e. g. *1*), in which the *Rollback* handler process for iDocs is located
- `jcoserver.idoc.rollback.process`: Path specification to an *X4* process serving as *Rollback* handler for iDocs

Connection settings to the SAP host
In the following properties the settings to connect to an SAP host are defined for the respective *JCo 2 Server* instance:

```
# connection String to SAP
jco.server.gwhost  = Gateway host
jco.server.gwserv  = Gateway service
jco.server.progid  = Program ID
jco.server.trace   = Boolean value
jco.server.unicode = Boolean value

# required for dynamic repository and metadata lookup
jco.client.client  = Client ID
jco.client.user    = User name
jco.client.passwd  = Password
jco.client.lang    = Client language
jco.client.sysnr   = SAP system number
jco.client.ashost  = SAP application server
jco.client.trace   = Boolean value
```

Explanation:

- `jco.server.gwhost`: Gateway host

- `jco.server.gwserv`: Gateway service (normally *sapgwNN*)
- `jco.server.progid`: Program ID of the SAP host
- `jco.server.trace`: Activate (*1*) or deactivate (*0*) RFC trace for the SAP Server
- `jco.server.unicode`: Connect (*1*) or do not connect (*0*) in Unicode mode
- `jco.client.client`: Client ID (three-digit numeric value)
- `jco.client.user`: User name
- `jco.client.passwd`: Password
- `jco.client.lang`: Language of the client
- `jco.client.sysnr`: SAP system number
- `jco.client.ashost`: SAP application server
- `jco.client.trace`: Activate (*1*) or deactivate (*0*) RFC trace

Function block/method definitions (X4 SAP JCo 2)

Within a configuration document (by default `<X4>/X4DB/0/jco_method_def.xml`) you can assign via the *JCo 2 Server* a corresponding X4 process for further processing for each function block (method), which is addressed by the SAP host.
Within the central configuration file `<X4>/X4config.xml` a function block/method definition must be referenced for each *JCo Server* instance.

Expected XML structure
Within the function block/method definition, you can create any number of `<method>` elements within the root element `<definitions>` by following this pattern:

```
<method>
  <name>Function block/method name</name>
    <wait_for_result>Boolean value</wait_for_result>
    <class>de.softproject.integration.sapjcoserver.requesthandlers.WorkflowStarter</
class>
    <properties>
      <property Name="workflowname">Path to the X4 process</property>
      <property Name="userID">User folder within the X4DB</property>
      <property Name="raw">Boolean value</property>
    </properties>
</method>
```

The following applies for each function block/method definition:

- The element `<name>` contains the name of the function block (method) called by SAP; The placeholder `*` represents all other function blocks, for which no explicit function block has been defined.
- Within the element `<wait_for_result>` you can configure whether SAP should wait for the result (*`false`*), or the function block/method call should be made asynchronously (*`true`*).
- The element `<class>` specifies the used class, which is stored for the processing of the function block or method (by default `de.softproject.integration.sapjcoserver.requesthandlers.WorkflowStarter` for

starting *X4* processes); Within the child elements of `<class>` you specify the corresponding properties in an element `<property>`.

For the *X4* process starter class, the following properties are specified:

- `workflowname`: Path to the *X4* process, which takes over the processing of the function block/ method call
- `userID`: Repository user folder containing the *X4* process
- `raw`: Process IDoc in raw format (*`true`*) or (*`false`*)

> **Compatibility notes**
> In older function block/method definitions you may find additional details within the elements `<type>` and `<property name = "languageID">`. They have no function and may be omitted therefore.

Example
In the following example the X4 process `save_idoc.wrf` is called for the function block (method) `ORDERS05` and data in the IDoc raw format is provided for the downstream processing.

```
<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>
```

Connection pooling (X4 SAP JCo 2)

The *SAP JCo 2 Server* can provide a pool for client connections for each instance. If no pool exists, a *JCo Client* connection will be made directly.
Setting for the Connection Pooling within the Server properties configuration file, see Properties for the JCo 2 server:

- `usePool:` Activate (*`true`*) or deactivate (*`false`*) Connection Poolin
- `poolSize:` Number of permitted client connections within the pool; Recommendation: Select a vaule corresponding at least to the sum of the *JCo 2 Server* instances (identified by a *ProgID*).

Pool size and number of SAP adapters
Increase the value specified within `poolsize` by the number of SAP adapters used within your X4 processes. For the SAP Connector (JCo 2), you can specify the used pool by using the adapter parameter `x4.connectionPoolUID` (arbitrary).

Updating the JCo 2 server configuration

To update configuration changes for the *SAP JCo Server*, it is not necessary to restart the complete X4 Server. Therefore, the X4 Server provides a JMX MBean `de.softproject.X4.X4Management` with the following three methods.

In *JBoss 4.2.3 GA*, you can call the JMX console in the browser via a URL following the pattern `http://host:port/jmx-console/`, e.g. *`http://localhost:8080/jmx-console/`*. There you will find the respective MBean `de.softproject.X4.X4Management`.

Restart *JCo Server*:	Call method `restartSAPJcoServer()` using **Invoke**
Shut down *JCo Server*:	Call method `stopSAPJcoServer()` using **Invoke**
Check *JCo Server* configuration regarding changes and read them:	Call method `reloadChangedSAPJCoServers()` using **Invoke**

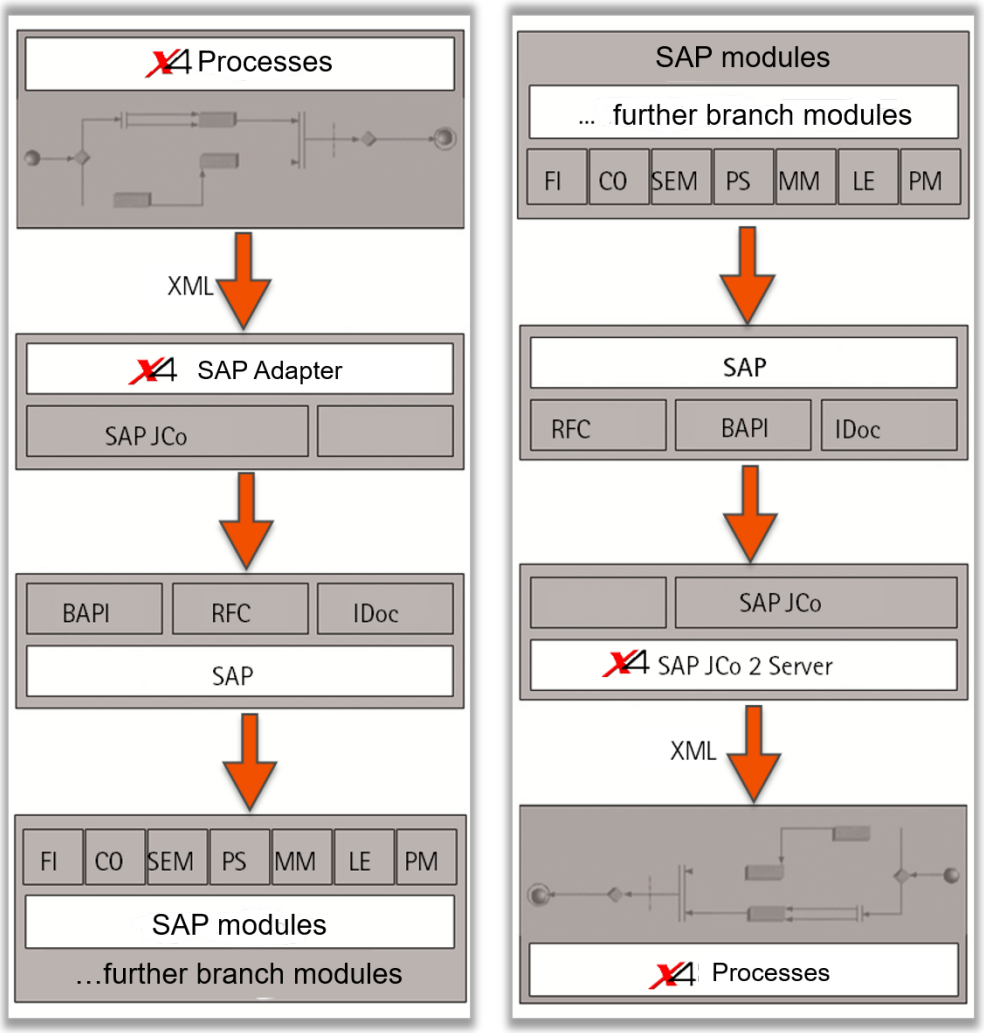
10.9.2.7 SAP Listener (JCo 3)

The *SAP Listener (JCo 3)* allows the processing of incoming communication requests via BAPI/RFC and of IDocs of an SAP system. If necessary, it automatically re-establishes broken connections. Former name until version 4.5.1: *X4 SAP JCo 3 Listener*

Functions

The *X4 Server* allows the bidirectional integration of *SAP R/3* systems using the *SAP Java Connector* in version 3.x (*JCo 3*). The following components are relevant:

- *SAP JCo 3 Server*: Allows the processing of incoming communication requests via BAPI/RFC and of IDocs of an SAP system. If necessary, it automatically re-establishes broken connections.
- *SAP Connector (JCo 3)*: In Technical Processes, this function adapter can invoke any Remote Function Call (RFC) or any BAPI function module in an SAP system. Moreover, IDocs can be sent to *SAP R/3*.
- *SAP Explorer (JCo 3)*: This adapter generates structure descriptions of IDocs and BAPIs, lists BAPIs or IDocs, or generates empty IDoc and BAPI structures. Depending on the operation, it outputs a corresponding XML document.



Multi-SAP Connectivity

Using the *JCo 3 Server*, multiple SAP systems can be connected via a respective separate *JCo 3 Server* instance. To use the *JCo 3 Server* a corresponding configuration is required:

- *X4* configuration within the `X4config.xml`; see SAP JCo 3 server configuration (X4config.xml)
- Definition of connection properties; see JCo3 Properties
- Assignment of the function block calls to Technical Processes and commit/rollback handlers; see Function block/method definitions (X4 SAP JCo 3)

Identification of JCo 3 Servers via ProgID

Each *JCo 3 Server* instance or any system connected to SAP communicates via a unique *ProgID*, which is configured within the server properties file, by default `jco_server_settings.prop` and is introduced to the SAP system at the start of the server instance.

Several *ProgIDs* can be used simultaneously for each SAP system. Using the *ProgID*, IDocs and RFC/BAPIs are "addressed" to the respective target system.

Installing X4 SAP JCo 3

Which steps are necessary to install *X4 SAP JCo 3*

> As an example, an installation for the JBoss application server based on a 32 bit Windows system will be described.

Prerequisites

To install an *SAP JCo 3 Server* within the *X4 Server* SAP libraries are needed, which are operating system and architecture-dependent. These libraries are also needed by the *SAP Connector (JCo 3)* and the *SAP Explorer (JCo 3)*.

As a registered SAP partner, you can download the necessary libraries and further documentation via *SAPNet* under http://service.sap.com/connectors.

> Make sure you select mutually compatible version numbers of the libraries (these are also operating system and architecture-dependent).
> Make sure that your operating system is compatible with the used JDK version and the version of your SAP system. The following matrix shows the relationship of these three variables.

		OS		SAP	
		32 bit	64 bit	32 bit	64 bit
JDK	32 bit	x	x	x	-
	64 bit	-	x	-	x

Installing the libraries

1. Copy the native libraries (depending on the system) to the respective folder of the application server for native binary dependencies or to the `bin` directory of Java.
   Example: Under Windows 32 bit: Copy `librfc32.dll`, `msvcr71.dll`, `msvcp71.dll` and `sapjcorfc.dll` to `C:\Windows\System32` or to `C:\X4\Server\jdk1.7\bin`.
2. Copy the following Java libraries to the classpath of the application server:
   - `sapidoc.jar`
   - `sapjco.jar`
   - `sapidocjco.jar`

   Example JBoss 4.2.3.GA: Copy these libraries to `C:\X4\JBoss4.2.3.GA\server\default\lib`.
   Example JBoss AS 7.1.1: Copy these libraries under `C:\X4\Server\jboss-as-7.1.1.Final\standalone\deployments` to the `lib` directory of `X4.ear`.
3. Restart the *X4 Server*.

Testing the installation

- Test the connection using the *SAP Explorer (JCo 3)* (analogous to the *X4 Experience* example `X4Experience/AdapterSuites/ X4SAPSuite// Processes/CreateEmptyBapi.wrf`)
- Make sure that the configuration of the *SAP JCo 3 Server* is complete
- Set the adapter parameter `SystemType` to *3* (for a connection to *SAP R/3*)
- Make sure that the adapter operation `Create Empty Bapi` is set
- Execute the process with the input XML `<Bapiname>BAPI's name</Bapiname>` and check the result XML

On this page:

- Prerequisites
- Installing the libraries
- Testing the installation

See also:

- Configuring X4 SAP JCo 3

Configuring X4 SAP JCo 3

Which configuration has to be made to start the *JCo 3 Server*, to make it accessible from the outside and to the assign the appropriate handler processes within the *X4 Server* to the incoming RFC/BAPIs and IDocs

SAP JCo 3 server configuration (X4config.xml)

How the *X4 SAP JCo 3 Server* is configured within the `X4config.xml`

ⓘ The main configuration file `<X4>\X4config.xml` doesn't contain any configuration options for the *X4 SAP JCo 3 Server* any more. Instead, the *JCo 3* configuration is located within properties files that are accessed via the adapter and/or the Listener configuration.

See also:

- JCo 3 properties
- JCo 3 Listener configuration
- Konfiguration über X4config.xml

JCo 3 Listener configuration

How to assign properties and methods to incoming *JCo 3* connections

The configuration of incoming *JCo 3* connections from one or several SAP systems is located within the file `saplistener.xml` in `<X4>\X4DB\0`. There, a *JCo 3* listener can be defined for each SAP system. The *JCo 3* Listener thereby references server properties and method definitions (in a specific XML document).

XML structure
To describe the structure of `saplistener.xml` also an XML schema definition `saplistener.xsd` is available within the same folder. This XSD is specified within the root element `<listeners>` of the Listener configuration.

```
<listener>
   <serverPropertiesURL>Path to the properties file</serverPropertiesURL>
   <methodDefinitionsURL>Path to the method definition</methodDefinitionsURL>
   <replaceEscapedControlChars>false</replaceEscapedControlChars><!-- true | false --
>
</listener>
```

Each element `<listener>` contains the following child elements:

- `<serverPropertiesURL>`: Valid URL to a properties text file that contains all connection settings for the SAP host
- `<methodDefinitionsURL>`: Valid URL to an XML document, in which all method definitions or function block calls and transaction handling processes are defined
- `<replaceEscapedControlChars>`: Replace masked control characters (e.g.: *&#31;*) within the XML strings returned by the JCo library by the replacement character ( ). Possible values are `true` and `false`. Do not replace masked control characters

Example
For the connection of two SAP hosts, two single listeners with separate properties and method definitions are defined. Moreover, with `replaceEscapedControlChars` *true* masked control characters (e.g.: `&#31;`) within the XML strings returned by the JCo library are replaced by the replacement character ( ).

```
<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>
```

On this page:

- XML structure
- Example

See also:

- JCo 3 properties
- Function block/method definitions (X4 SAP JCo 3)

JCo 3 properties

How to define the connection parameters for the communication between client and host

The connection data for the communication between client and host can be defined globally in a properties file. For incoming connections, such a properties file is specified for each *X4 SAP JCo 3* listener in `<X4> /X4DB/0/saplistener.xml`.

For outgoing connections via *SAP Connector (JCo 3)* and *SAP Explorer (JCo 3)*, specify the path to such a properties file within the adapter parameter `property file`.

Parameter values set within the adapter overwrite the configuration in the properties file at runtime. All properties that are set in the properties file (also additional properties), are taken over by the *X4 SAP JCo 3* system.

Properties structure

```
# connection String to SAP
jco.server.gwhost  = Gateway Host
jco.server.gwserv  = Gateway Service (normally sapgwNN)
jco.server.progid  = Program ID
jco.server.trace   = Activate (1) or deactivate (0) RFC Trace
jco.server.unicode = Connect (1) or do not connect (0) in Unicode mode

# equivalent to (dynamic) adapter parameter 'User'
jco.client.user = User name

# equivalent to (dynamic) adapter parameter 'Password'
jco.client.passwd = Password

# equivalent to (dynamic) adapter parameter 'Client'
jco.client.client = SAP client (3-digit numeric value)

# equivalent to (dynamic) adapter parameter 'Language'
jco.client.lang = Login language

# equivalent to (dynamic) adapter parameter 'SystemNumber'
jco.client.sysnr = SAP system number

# 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 = Type of the remote hosts

# set to true if transactions are used
# equivalent to (dynamic) adapter parameter 'TransactionEnabled'
x4.transactionEnabled = Enable (true) or disable (false) transactionale processing

# Should be set for each initiated transaction
# equivalent to (dynamic) adapter parameter 'TransactionId'
x4.transactionID = Transaction ID (for each initiated transaction)
```

Example

In the following an example of a properties file with some values set:

```
# 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
```

On this page:

- Properties structure
- Example

See also:

- JCo 3 Listener configuration
- Function block/method definitions (X4 SAP JCo 3)

Function block/method definitions (X4 SAP JCo 3)

How to assign an X4 process to RFC/BAPIs within the SAP listener

Within a specific XML document (e. g. `<X4>/X4DB/0/jco_method_def.xml`) you can assign via the *JCo 3 Server* a corresponding X4 process for further processing for each function block (method), which is addressed by the SAP host.

XML structure for the SAP listener configuration

The SAP listener configuration file `<X4>/X4DB/0/saplistener.xml` references for each *JCo 3 Server* instance an XML document with function block/method definitions. Here, you can create any number of `<method>` elements within the root element `<definitions>` by following this pattern:

```
<method>
  <name>Function block/method name</name>
    <wait_for_result>Boolean value</wait_for_result>
    <class>de.softproject.integration.sapjcoserver.requesthandlers.
           WorkflowStarter</class>
    <properties>
      <property name="workflowname">Path to the X4 process</property>
      <property name="userID">User folder within the X4DB</property>
      <property name="raw">Boolean value</property>
    </properties>
</method>
```

The following applies for each function block/method definition:

- The element `<name>` contains the name of the function block (method) called by SAP; The placeholder `*` represents all other function blocks, for which no explicit function block has been defined. Moreover, transaction handlers can be defined here (see below).
- Within the element `<wait_for_result>` you can configure whether SAP should wait for the result (*`false`*), or the function block/method call should be made asynchronously (*`true`*).
- The element `<class>` specifies the used class, which is stored for the processing of the function block or method (by default `de.softproject.integration.sapjcoserver.requesthandlers.WorkflowStarter` for starting *X4* processes); Within the child elements of `<class>` you specify the corresponding properties in an element `<property>`.

For the *X4* process starter class, the following properties are specified:

- `workflowname`: Path to the *X4* process, which takes over the processing of the function block/method call
- `userID`: Repository user folder containing the *X4* process
- `raw`: Process IDoc in raw format (*`true`*) or (*`false`*)

Compatibility notes
In older function block/method definitions you may find additional details within the elements `<type>` and `<property name = "languageID">`. They have no function and may be omitted therefore.

> When migrating to JCo 3, please note that the attribute `name` within the element `<property>` was capitalized under JCo 2. However, under JCo 3 the attribute is lowercase.

Transaction handler for IDocs and BAPI calls
In *X4 SAP JCo 3*, the transaction handling has been integrated within the method definition. For processing incoming IDocs and BAPI calls, you can define a *commit* handler and a *rollback* handler and

assign them to an X4 process by specifying within one of the method definitions the following values in the element `<name>`:

- *`IDOC+COMMIT`*: End IDoc transaction (*Commit*)
- *`IDOC+ROLLBACK`*: Cancel and reset IDoc transaction (*Rollback*)
- *`BAPI+COMMIT`*: End BAPI transaction (*Commit*)
- *`BAPI+ROLLBACK`*: Cancel and reset BAPI transaction (*Rollback*)

Global handler
If no specific handler applies, you can define a global handler, by entering an asterisk (`*`) within the element `<name>` in the method definition.

Example
In the following example the X4 process `save_idoc.wrf` is called for the function block (method) `ORDERS05` and data in the IDoc raw format is provided for the downstream processing. In addition, a BAPI commit handler and a global handler are defined without an assigned X4 process.

```
<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>
```

On this page:

- XML structure for the SAP listener configuration
- Compatibility notes
- Transaction handler for IDocs and BAPI calls
- Global handler
- Example

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`: Reads in a text document and outputs an XML document with unchanged character encoding, enclosing the content of the text document with a CDATA area.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`WrapData`*: Convert text document into an XML document with CDATA area.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *en.softproject.integration.adapter.ASCII2XML*: Main class (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.10.1.2 ASCII Writer

`ASCII Writer`: Converts an XML document according to the rules of a descriptor file into a structured text document with unmodified encoding.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** *Execute*: Write an ASCII document according to the rules of a descriptor file specified within the parameter `DescFile`

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.asciiwriter.ASCIIWriter*: Main class (Default)
DescFile	Name of the descriptor file describing the text document's data structure. The descriptor file should be located in the same X4 Repository user directory (e.g. `X4DB/1/Project/Descr.ad`) **Possible values:** Path and file name (e.g. */Project/Descr.ad*)

State values

-1	An error occurred during the adapter's execution (see X4 Server log).
0	The adapter has been executed successfully.

10.10.1.3 Text to Lines Converter

`Text to Lines Converter`: Reads all lines or specified lines of a text document and outputs an XML document containing an XML element for each line.

As input, the adapter expects a text document provided by the preceding process action. The `Iterate` operation allows to iterate through the input text document and to output a result document for each iteration within the Technical Process.

> **Please note:**
>
> - If you want to define multiple *Text to Lines Converters* within the same process context, set in parameter `id` a unique ID for each adapter, e.g. *`1001`*, *`1002`* etc. This ensures that each adapter will iterate or will be reset correctly (see operation `Clear`).
> - When the `id` parameter's value remains *`0`*, the number of the process action (Action ID) will be used as ID in the current process. This ID might not be unique if a subprocess contains a *Text to Lines Converter* with the same Action ID.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • *`Parse`*: Read lines of the text document • *`Iterate`*: Activate iterator mode and read the text document line by line • *`Clear`*: Reset iterator of a *Text to Lines Converter* having the same ID as the corresponding *Text to Lines Converter* with the operation `Iterate` (see parameter `id`)

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.adapter.text2lines.TextToLinesAdapter`*: Main class (Default)
start	First line to be read (counting begins with `0`) **Possible values:** • Any integer number • *`0`*: Start to read from the first line (default)

count	Number of lines to be read (when using operation `Iterate` for each iteration) **Possible values:** • Any integer number • *0*: No limitation, read all lines at once (default)
id	Unique ID of the *Text to Lines Converter* component within the process context respectively the process instance (if you want to use multiple independently iterating *Text to Lines Converters* within the same process context) **Possible values:** • *0*: Use current action ID (process action number) as ID; Note, that another *Text to Lines Converter* in a subprocess could have the same action ID. (default) • Any integer number except *0* to uniquely identify the corresponding *Text to Lines Converter* component within the process context; Use a value *>1000* to avoid catching an already existing action ID accidentally.

Output

The adapter outputs a UTF-8-encoded XML document with the following structure. An XML element `<Line/>` containing the line number and the line's content will be created for each line:

```
<Text>
  <Line lineNo="line number"><!--line content--></Line>
</Text>
```

State values

-1	Error while executing the adapter (for details, see server log)
0	The adapter doesn't output data (anymore). When using the operation `Iterate`, this status can be used as stop criterion of a loop within the Technical Process and it indicates that no further data row is available.
1	The adapter outputs a result document. When using the operation `Iterate`, this status can be used as condition to enter a loop within the Technical Process and it indicates that another data row is available.

10.10.1.4 Text to Lines Converter from URL

`Text to Lines Converter from URL`: Reads lines of a text document from any valid URL and outputs an XML document containing an XML element for each line. With the operation Iterate, the adapter runs through the input document and outputs a result document for each iteration within the Technical Process.

> **Please note:**
>
> - If you want to define multiple *Text to Lines Converters from URL* within the same process context, set in parameter `id` a unique ID for each adapter, e.g. *`1001`*, *`1002`* etc. This ensures that each adapter will iterate or will be reset correctly (see operation `Clear`).
> - When the `id` parameter's value remains *`0`*, the number of the process action (Action ID) will be used as ID in the current technical process. This ID might not be unique if a subprocess contains a *Text to Lines Converter* with the same Action ID.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *`Parse`*: Read the complete text document • *`Iterate`*: Activate iterator mode and read the text document line by line • *`Clear`*: Reset iterator of a *Text to Lines Converter from URL* having the same ID as the corresponding *Text to Lines Converter from URL* with the operation *`Parse`* (see parameter `id`)

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.adapter.text2lines.TextToLinesUrlAdapter`*: Main class (Default)
start	First line to be read (counting begins with 0) **Possible values:** • Any integer number • *`0`*: Start to read from the first line (Default)

count	Number of lines to be read (when using operation `Iterate` for each iteration) **Possible values:** • Any integer number • *0*: No limitation, read all lines at once (Default)
id	Unique ID of the *Text to Lines Converter from URL* component within the process context respectively the process instance (if you want to use multiple independently iterating *Text to Lines Converters from URL* within the same process context) **Possible values:** • *0*: Use current action ID (process action number) as ID; Note, that another *Text to Lines Converter from URL* in a subprocess could have the same action ID. (Default) • Any integer number except *0* to uniquely identify the corresponding *Text to Lines Converter from URL* component within the process context; Recommendation: use a value *>1000* to avoid catching an already existing action ID accidentally.
url	URL of the text document to be read **Possible values:** Any valid URL (e.g. *xstore://Project/Folder/File.txt*)
encoding	Encoding of the input text document **Possible values:** • Any valid encoding (e.g. *UTF-8*) • *(none)*: Use default system encoding

Output

The adapter outputs a UTF-8-encoded XML document with the following structure. An XML element `<Line/>` containing the line number and the line's content will be created for each line:

```
<Text>
    <Line lineNo="line number"><!--line content--></Line>
</Text>
```

State values

-1	Error while executing the adapter (for details, see server log)

0	The adapter doesn't output data (anymore). When using the operation `Iterate`, this status can be used as stop criterion of a loop within the Technical Process and it indicates that no further data row is available.
1	The adapter outputs a result document. When using the operation `Iterate`, this status can be used as condition to enter a loop within the Technical Process and it indicates that another data row is available.

10.10.2 Base64 Batch Decoder

`Base64 Batch Decoder`: Decodes Base64-encoded data from an adapter-specific XML structure, splits the result into separate files and stores them in a folder.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Decode`*: Write data from the XML document to Base64 encoded files.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.docadapter.DocAdapter`*: Main class (default)
directory	Destination folder where the decoded files should be stored **Possible values:** Any valid string (e.g. *`%USER_COLLECTION%\Project\batch\`*)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

This adapter expects an adapter-specific input XML structure. This contains the file name and the Base64-encoded file content for each file to be decoded. Each `<document/>` element represents a file to be decoded.

```
<data>
    <documents>
        <document>
            <filename><!--Dateiname--></filename>
            <content><!--Base64-kodierte Daten--></content>
        </document>
    </documents>
</data>
```

10.10.3 Base64 Converter

`Base64 Converter`: Encodes or decodes data in Base64 encoding. When decoding, the adapter distinguishes based on the MIME type: if the MIME type is `text/xml` or `application/xml`, then the input document is interpreted as an XML document and the text content of the root element is decoded, otherwise the complete document is interpreted as a Base64 encoded string.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *Encode*: Base64 encode data • *Decode*: Decode data in Base64 encoding (depending on MIME type).

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.Base64Adapter*: Main class (default)
`chunk`	Divide output into blocks with a length of 76 characters **Possible values:** • *true*: Divide string into blocks of lines • *false*: Do not divide string (default)
`encoding`	Used character encoding **Possible values:** String with the used character encoding, e.g. `UTF-8` (Standard), `windows-1251`, `ISO 8859-1`, ...

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.10.4 Charset Changer

`Charset Changer`: Changes the character encoding of a document.

> The characters of the input document are not converted into the target character encoding! Use the Adapter *Encoding Converter* for this, see Encoding Converter.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Execute`*: Change character set of file

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *en.softproject.integration.adapter.CharsetChanger*: Main class (default)
InCharset	Character set of the input document **Possible values:** • Any character encoding known in Java (e.g. *Windows-1252*) • *`UTF-8`*: Unicode character encoding UTF-8 (standard)
OutCharset	Character set of the output document **Possible values:** • Any character encoding known in Java (e.g. *`ISO-8859-1`*) • *`UTF-8`*: Unicode character encoding UTF-8

Status values

`-1`	The operation was executed successfully.
`1`	The operation failed due to a technical error.

10.10.5 CSV

10.10.5.1 CSV Converter

`CSV Converter`: Reads a CSV text document (*comma separated values*) from the previous process action and outputs an XML document with UTF-8 character encoding and with element names that can refer to the column names.

Lines before the actual beginning of CSV data rows can be omitted as well as comment lines. Both CR (*carriage return*) and CRLF (*carriage return line feed*) will be interpreted as line breaks. Using the operation `Iterate`, the adapter can iterate through the input document and output a result document for each iteration within the *X4* process.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *`Parse`* : Read CSV text document in one step • *`Iterate`* : Read CSV text document iterative • *`Clear`* : Reset the iterator of a *CSV Converter* having the same ID as the corresponding *CSV Converter* with the operation *`Iterate`* (see parameter `id`)

Parameters

`Adapter`	Main adapter class (Do not change!) **Possible values:** *de.softproject.integration.adapter.csv.CsvAdapter*: Main class (default)
`start`	First data row to be read (counting starts with 0; if the CSV file contains a header row, it will be omitted) **Possible values:** • Positive integer • *0*: Read from the first line (Default)

`count`	Number of rows to be read (with operation Iterate for each iteration) **Possible values:** • Positive integer • *0*: No limitation, e.g. read all rows at once (default)
`id`	Unique ID of the *CSV Converter* component within the process context, respectively the process instance (if you want to use multiple independently iterating *CSV Converters* within the same process context) **Possible values:** • *0* : Use current action ID (process action number) as ID; Please note that another *CSV Converter* in a subprocess could have the same action ID. (Default) • Any integer number except *0* to uniquely identify the corresponding *CSV Converter* component within the process context; Recommendation: use a value *>1000* to avoid catching an already existing action ID accidentally.
`columnNamesInFirstRow`	Interpret data fields in the CSV text file's first row as column headers and name the data field elements within the XML document correspondingly, if the values are valid XML names **Possible values:** • *Enabled*: The first row of the CSV text document contains column headers that are applicable as valid XML element names (without namespace declarations). • *Disabled*: The CSV text document contains only data rows; data elements will be named <Column> or <Column_n> (if the parameter `useIndexedColumns` has been enabled). (Default)

`normalizeHeaderNames`	Defines how headers should be normalized if they contain characters that are not XML-compliant, such as spaces or special characters. **Possible values:** • *Special element name* (default): Do not normalize header name, but replace it with `<Column>` (Ex: `author book` → `<column>`) • *Skip non compliant characters*: Skip non-XML characters (Ex: `author book` → `<authorbook>`) • *Replace with dash*: Replace non-XML characters with – (Ex: `author book` → `<author-book>`) • *Replace with underscore*: Replace non-XM characters with _ (Ex: `author book` → `<author_book>`)
`columnNamesAfterStart`	Interpret values from the row defined within the parameter `start` as column headers, and name the data field elements within the XML document correspondingly, if the values are valid XML names **Possible values:** • *Enabled*: Ignore skipped rows and use column headers from the first processed row • *Disabled*: The first processed row doesn't contain column headers; data elements will be named <Column> or <Column_n> (if the parameter `useIndexedColumns` has been enabled). (Default)
`delimiter`	Delimiter character between the fields of a record in the CSV text document **Possible values:** • Any character (no escape sequences such as `\t`) • `;`: Semicolon (default)
`rowName`	Element name in the output XML document that wraps each data row **Possible values:** XML element name (without namespace)

`excel`	Defines if the processed CSV text document has been exported from *MS Excel* (not a standard CSV format) and if the adapter must convert it **Possible values:** • *Enabled*: Process a CSV text document in the *MS Excel* CSV format • *Disabled:* Process a standard CSV text document (Default)
`useIndexedColumns`	Add column numbers (starting with 0) to data element names, if no column headers are available **Possible values:** • *Enabled:* Append actual column number to the data element name (e.g. *<Column_0>*) • *Disabled:* All data elements will be named `<Column>` (Default)
`commentStart`	Prefix string to indicate comment lines; Matching lines will be omitted. **Possible values:** • Any string (except delimiter or quoting characters) (e.g. ******) • *(empty)*: No comment handling; Lines will be interpreted as data rows. (Default)
`inputEncoding`	Defines which character encoding is used in the adapter input. **Possible values:** • `UTF-8` • `windows-1252` • `ISO 8859-1` ⓘ For the adapter output, always UTF-8 is used as character encoding.

Status values

-1	An error occurred during the adapter's execution (for details see server log).

0	The adapter doesn't output data (anymore). When using the operation `Iterate`, this status can be used as stop criterion of a loop within the *X4* process and it indicates that no further data row is available.
1	The adapter outputs a result document. When using the operation `Iterate`, this status can be used as condition to enter a loop within the *X4* process and it indicates that another data row is available.

Input

The adapter expects any structured CSV text file (that might be generated by *MS Excel*) with any delimiter character and with optional comment lines.

Output

This adapter outputs a UTF-8-encoded XML document with the following structure. For each processed line an element will be generated that has a name specified within the parameter `rowName`:

```
<Csv>
    <!-- If column names are copied -->
    <Elementname rowNo="row number in the input">
        <Columnname>value</Columnname>
        <!--...-->
    </Elementname>
    <!-- If no column names are available -->
    <Elementname rowNo="column number in the input">
        <Column>value</Column>
        <!--...-->
    </Elementname>
    <!-- If no column names are available and parameter 'useIndexedColumns' is
enabled -->
    <Elementname rowNo="row number in the output">
        <Column_Columnnumber>value</Column_Columnnumber>
        <!--...-->
    </Elementname>
</Csv>
```

Example

A CSV text file with comments is processed that contains column headers only in the second line, for instance:

```
**** 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","Sa
les","Obere Str. 57","12209" **** This line is just a comment and should not be
processed. **** "ANTON","Moreno Taquería","Antonio Moreno","Owner","Mataderos 2312","
05023"
```

This requires to enable the adapter parameter `columnNamesAfterStart`, and to set the parameter `start` to *`1`* because both comment line and column header line won't be counted. When using the value *`Dataset`* for the parameter `rowName`, the following XML document will be generated:

```
<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`: Reads a CSV text document (*comma separated values*) from any URL and outputs an XML document with UTF-8 character encoding and with element names that can refer to the column names.
Lines before the actual beginning of CSV data rows can be omitted as well as comment lines. Both CR (*carriage return*) and CRLF (*carriage return line feed*) will be interpreted as line breaks. Using the operation `Iterate`, the adapter can iterate through the input document and output a result document for each iteration within the Technical Process.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *`Parse`*: Read CSV text document in one step • *`Iterate`*: Read CSV text document iterative • *`Clear`*: Reset the iterator of a *CSV Converter form URL* having the same ID as the corresponding *CSV Converter from URL* with the operation *`Iterate`* (see parameter `id`)

Parameters

`Adapter`	Main adapter class (Do not change!) **Possible values:** *de.softproject.integration.adapter.csv.CsvAdapter*: Main class (default)
`start`	First data row to be read (counting starts with 0; if the CSV file contains a header row, it will be omitted) **Possible values:** • Positive integer • *0*: Read from the first line (Default)
`count`	Number of rows to be read (with operation `Iterate` for each iteration) **Possible values:** • Positive integer • *0*: No limitation, e.g. read all rows at once (default)
`id`	Unique ID of the *CSV Converter from URL* component within the process context or the process instance (if you want to use multiple independently iterating *CSV Converters from URL* within the same process context) **Possible values:** • *0*: Use current action ID (process action number) as ID; Please note that another *CSV Converter from URL* in a subprocess could have the same action ID. (Default) • Any integer number except *0* to uniquely identify the corresponding *CSV Converter from URL* component within the process context; Recommendation: use a value *>1000* to avoid catching an already existing action ID accidentally.

`columnNamesInFirstRow`	Interpret data fields in the CSV text file's first row as column headers and name the data field elements within the XML document correspondingly, if the values are valid XML names **Possible values:** • *Enabled*: The first row of the CSV text document contains column headers that are applicable as valid XML element names (without namespace declarations). • *Disabled*: The CSV text document contains only data rows; data elements will be named `<Column>` or `<Column_n>` (if the parameter `useIndexedColumns` has been enabled). (Default)
`normalizeHeaderNames`	Defines how headers should be normalized if they contain characters that are not XML-compliant, such as spaces or special characters. **Possible values:** • *Special element name* (default): Do not normalize header name, but replace it with `<Column>` (Ex: `author book` → `<column>`) • *Skip non compliant characters*: Skip non-XML characters (Ex: `author book` → `<authorbook>`) • *Replace with dash*: Replace non-XML characters with - (Ex: `author book` → `<author-book>`) • *Replace with underscore*: Replace non-XML characters with _ (Ex: `author book` → `<author_book>`)
`columnNamesAfterStart`	Interpret values from the row defined within the parameter `start` as column headers, and name the data field elements within the XML document correspondingly, if the values are valid XML names **Possible values:** • *Enabled*: Ignore skipped rows and use column headers from the first processed row • *Disabled*: The first processed row doesn't contain column headers; data elements will be named `<Column>` or `<Column_n>` (if the parameter `useIndexedColumns` has been enabled). (Default)

`delimiter`	Delimiter character between the fields of a record in the CSV text document **Possible values:** • Any character (no escape sequences such as \t) • *;*: Semicolon (default)
`rowName`	Element name in the output XML document that wraps each data row **Possible values:** XML element name (without namespace)
`excel`	Defines if the processed CSV text document has been exported from *MS Excel* (not a standard CSV format) and if the adapter must convert it **Possible values:** • *Enabled*: Process a CSV text document in the *MS Excel* CSV format • *Disabled:* Process a standard CSV text document (Default)
`useIndexedColumns`	Add column numbers (starting with 0) to data element names, if no column headers are available **Possible values:** • *Enabled*: Append actual column number to the data element name (e.g. *<Column_0>*) • *Disabled:* All data elements will be named `<Column>` (Default)
`commentStart`	Prefix string to indicate comment lines; Matching lines will be omitted. **Possible values:** • Any string (except delimiter or quoting characters) (e.g. ****) • *(empty)*: No comment handling; Lines will be interpreted as data rows. (Default)

`inputEncoding`	Character encoding of the input **Possible values:** • Any valid character encoding (e.g. *`UTF-8`*) • *`(empty)`*: Use default encoding of the runtime environment
`url`	URL of the CSV text document to be read **Possible values:** Valid URL (e.g. *`xstore://Project/Folder/File.csv`*)
`encoding`	Character encoding of the CSV text document **Possible values:** • Any valid character encoding (e.g. *`UTF-8`*) • *`(empty)`*: Use default encoding of the runtime environment

Status values

-1	An error occurred during the adapter's execution (for details see server log).
0	The adapter doesn't output data (anymore). When using the operation `Iterate`, this status can be used as stop criterion of a loop within the Technical Process and it indicates that no further data row is available.
1	The adapter outputs a result document. When using the operation `Iterate`, this status can be used as condition to enter a loop within the Technical Process and it indicates that another data row is available.

Input

The adapter expects any structured CSV text file (that might be generated by *MS Excel*) with any delimiter character and with optional comment lines.

Output

This adapter outputs a UTF-8-encoded XML document with the following structure. For each processed line an element will be generated that has a name specified within the parameter `rowName`:

```
<Csv>
    <!-- If column names are copied -->
    <Elementname rowNo="row number in the input">
        <Columnname>value</Columnname>
        <!--...-->
    </Elementname>
    <!-- If no column names are available -->
    <Elementname rowNo="column number in the input">
        <Column>value</Column>
        <!--...-->
    </Elementname>
    <!-- If no column names are available and parameter 'useIndexedColumns' is
enabled -->
    <Elementname rowNo="row number in the output">
        <Column_Columnnumber>value</Column_Columnnumber>
        <!--...-->
    </Elementname>
</Csv>
```

Example

A CSV text file with comments is processed that contains column headers only in the second line, for instance:

```
**** 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","Sa
les","Obere Str. 57","12209" **** This line is just a comment and should not be
processed. **** "ANTON","Moreno Taquería","Antonio Moreno","Owner","Mataderos 2312","
05023"
```

This requires to enable the adapter parameter `columnNamesAfterStart`, and to set the parameter `start` to *`1`* because both comment line and column header line won't be counted. When using the value *`Dataset`* for the parameter `rowName`, the following XML document will be generated:

```
<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`: Converts an XML file into a CSV file.

Properties

Operation	Defines the operation executed by the adapter. **Possible values:** • *`Convert`*: Converts the XML file into a CSV file.

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.xml2csv.XML2CSVAdapter*: Main class (default)
`headerInOutput`	*Required*. Defines if there is a header in the CSV file to be created. **Possible values:** `true` (default) / `false`
`delimiter`	*Required*. Defines the delimiter in the CSV file to be created. **Possible values:** Any delimiter, e.g. `;`, `,` (default) or `\t` (tab as delimiter)

`lineEnd`	*Required*. Defines the character that marks the end of line in the CSV file to be created. **Possible values:** Character for the end of line, e.g. `\n` (default Unix), `\r\n` (defaultMS Windows)
`quotechar`	Defines the quotes for elements within the CSV file to be created. **Possible values:** • Any quote character, e.g. " • Default: Not set
`escapechar`	Defines the character that escapes certain characters that would otherwise be interpreted as control characters. **Possible values:** • Any character, e.g. " • Default: Not set
`encoding`	Defines the character encoding for the operation. **Possible values:** • Any character encoding, e.g. `UTF-8` • Default: `UTF-8`

Status values

`1`	The operation was executed successfully.
`-1`	The operation could not be executed.

Input

The adapter expects the following input structure:

Input Structure

```
<?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>
```

If the parameter `headerInOutput` is set to *true*, a header is created in the output CSV file. The columns names are defined with the `name` attribute within the `Header/Column` element. As many columns are created as there are `name` attributes.
If the parameter `headerInOutput` is set to *false*, no header is created in the output CSV file. The number of columns created is the maximum number of `Column` elements within a `Record` element.
Each entry must be contained in a `Record` element. If a header is defined, the values are assigned to the column with the corresponding `name`.

Output

The adapter outputs a CSV document with the structure that is defined in the input document.

Example

Example 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>
```

The adapter parameters are set as follows:

- `headerInOutput`: *true*
- `delimiter`: *,*
- `encoding`: *UTF-8*

The above input leads to the following output:

Example 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`: Converts the character encoding of an XML document, and changes both the character encoding declaration (in the XML prologue) and the actual character encoding of the content.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** *Execute*: Convert characters of the input XML document into the character encoding specified within parameter `TargetEncoding`

Parameter

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.encodingconversion.DocumentEncodingAdapter*: Main class (Default)
TargetEncoding	New character encoding of the output XML document **Possible values:** Any character encoding supported by the Java runtime environment (e.g. *ISO-8859-1*)

10.10.7 Hex Converter

`Hex Converter`: Converts binary data to hexadecimal and decodes hexadecimal encoded data.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *Encode*: Encode data hexadecimal • *Decode*: Decode hexadecimal data

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.HexAdapter*: Main class (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.10.8 Image Processing

10.10.8.1 Image to Image Converter

? Unknown Attachment `Image to Image Converter`: Converts and scales an image file to another image format. The following formats can be read and created: BMP, GIF, JPEG, JPEG 2000, PNG, RAW, TIFF and WBMP. The image files are specified in an adapter-specific input XML document. The adapter offers the possibility of specifying formatting options during conversion.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Transformation`* : Convert image format

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.image2image.ImageToImageAdapter`*: Main class (default)
width	Width of the output image in pixels (ignored if `maxSize` parameter is set). **Possible values:** Any integer
height	Height of the output image in pixels (ignored if `maxSize` parameter is set). **Possible values:** Any integer
maxSize	If the image is to be scaled in the original aspect ratio: Maximum width or height of the output image in pixels (`width` and `height` parameters are ignored). **Possible values:** • Any integer • *0*: *Do not* keep original aspect ratio and use width and height parameters for scaling (default).

scaleAlgorithm	Scaling algorithm used **Possible values:** • *SCALE_DEFAULT*: Use standard algorithm • *SCALE_FAST*: Use fast algorithm • *SCALE_SMOOTH*: Use edge smoothing algorithm • *SCALE_REPLICATE*: Use pixel repeat algorithm • *SCALE_AREA-AVERAGING*: Use area averaging algorithm
outputFormat	Output image format **Possible values:** • *BMP*: Create Windows Bitmap • *GIF*: Create GIF compressed image • *JPEG*: Create JPEG-compressed image • *JPEG 2000:* Create image in *JPEG 2000* format • *PNG*: Create PNG image • *RAW*: Create raw image • *TIFF*: Create image in TIF format • *WBMP*: Create image in *Wireless Bitmap* format

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.10.8.2 Multipage TIFF Creator

? Unknown Attachment `Multipage TIFF Creator`: Creates a *Multipage Tagged Image File Format* (TIFF) document from image files of type BMP, GIF, FlashPix, JPEG, PNG, PNM, TIFF and WBPM. The image files are specified in an adapter-specific input XML document. The adapter outputs a multi-page TIFF image as a binary data stream with a maximum size of 4 GB.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *CreateMultiTiff*: Create multipage TIFF document

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.tiff.Img2Tiff*: Main class (default)
compression	Compression type **Possible values:** • *Deflate*: Compress lossless via deflate • *JPEG*: Use JPEG compression (lossy) • *None*: Do not compress (default) • *Packbits*: Compress losslessly via PackBits run-length encoding.
deflateLvl	Deflate level (if *Deflate* compression type is selected and *Deflate* compression is to be applied to the entire TIFF document). **Possible values:** • Any integer from *1–9* (compresses fastest at *1*, most verbose at *9*). • *0*: Do not apply *Deflate* compression to the entire TIFF (default).
resX	Horizontal screen/print resolution value **Possible values:** • Any integer (e.g. *72*) • *0*: Do not set any resolution value
resY	Vertical screen/print resolution value **Possible values:** • Any integer (e.g. *72*) • *0*: Do not set any resolution value

resUnit	Unit for screen/print resolution **Possible values:** • *`No absolute unit`*: Do not set an absolute unit • *`Inch`*: Use inch as unit (DPI or PPI) • *`Centimeter`*: Use centimetre as unit

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

This adapter expects an adapter-specific input XML structure in which all image files to be processed are referenced via a valid URL protocol. You can use an optional thumbnail_scale attribute to specify whether a scaled version of the image should also be inserted as a page in the TIFF file. The attribute value specifies the scaling factor.

```
<imagechooser>
   <image url="URL zur Bilddatei" thumbnail_scale="Skalierungsfaktor" />
</imagechooser>
```

Example

The following input XML structure references two images, one via `xstore://` from the X4 repository, the second via an absolute path specification with `file://`. The function adapter creates a three-page TIFF file that contains the first image unchanged, the version of the first image reduced by 50% on the second page, and the second image on the third page.

```
<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`: Identifies the properties of a TIFF graphic. This adapter expects a TIFF graphic as input and outputs all identified TIFF properties (dimensions, number of pages, bit depth, orientation etc.) as XML document.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** *ReadFromImage*: Read properties from a TIFF graphic

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.tiff.TIFFpropertie s*: Main class (Default)

10.10.9 Java Properties File Reader

`Java Properties File Reader`: Reads a Java properties text file and stores all or selected properties in a hash table that is globally available to all Technical Processes during server runtime. Each property is assigned a unique key. The adapter can also read the global hash table and output its contents in an XML document either completely or after a regex filter.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *ReadFromFile*: Read properties text file and store properties completely or with a RegEx filter in a global hash table. • *ReadFromMap*: Read global hash table and output its contents in an XML document

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration. adapter.properties.PropertiesReader*: Main class (default)
file	Path to the properties text file in the repository or absolute path in the file system of the *X4 server* **Possible values:** • Any repository path (e.g. *%USER_COLLECTION%/Project/ Folder/sample.properties*) • Any absolute path (e.g. *C:\Data\sample.properties*)

fileKey	Key that is prefixed to the properties read in to make them uniquely identifiable. For example, a property sample.property is supplemented with the key `x4propkey` and stored in the hash table as `x4propkey.sample.property`. **Possible values:** Any string (e.g. *x4propkey*).
filterRegex	Regular expression by which the properties are to be filtered (only for `ReadFromFile` operation); multiple filter expressions can be specified separated by semicolons (`;`). **Possible values:** Any regular expression (e.g. `.*sample.*`)

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

10.10.10 QR Code Generator

`QR Code Adapter`: Generates and reads QR codes.

Properties:

`Operation`	Defines the operation executed by the adapter. **Possible values:** • `Consume`: Reads a QR code. • `Generate`: Generates a QR code.

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.qrcode.QRCodeGenerator*: Main class (default)
`imageWidth`	*Required*. Width of the QR code to be generated in pixels. **Possible values:** Any integer (default: `350`)
`imageHeight`	*Required*. Height of the QR code to be generated in pixels. **Possible values:** Any integer (default: `350`)

`imageFormat`	*Required*. Data format of the graphics file containing the QR code to be generated. **Possible values:** • `JPEG 2000` • `PNG` (default) • `RAW` • `TIFF` • `WBMP`

Status values:

`1`	The operation was executed successfully.
`-1`	An error occurred during the execution.

Input

The adapter expects different data as input depending on the operation to be executed.

- `Consume` operation: Graphics file with QR code
- `Generate` operation: Raw text

Output:

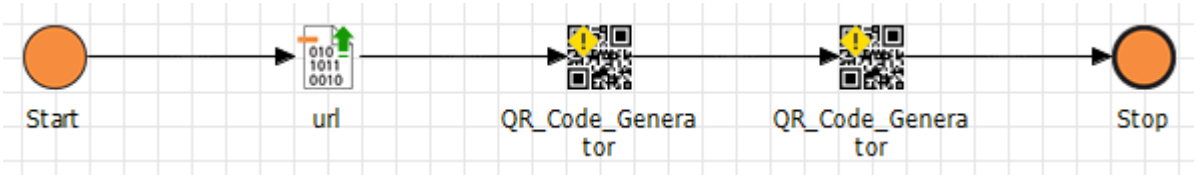
The adapter provides different data as output depending on the executed operation.

- `Consume` operation: Content that was read from the QR code.
- `Generate` operation: Graphics file with QR code

Example:

With the QR Code Generator, QR codes that contain a URL can be created and read.

The process then looks as follows:



Label	File	Description
url	url.txt	Text file that contains the URL `https://softproject.de/de/`.
QR_Code_Generator	Adapter QR Code Generator	Operation `Generate`
QR_Code_Generator	Adapter QR Code Generator	Operation `Consume`

Parameter of the adapter:

- `imageWidth`: 350
- `imageHeight`: 350
- `imageFormat`: PNG

10.10.11 Regular Expression

? Unknown Attachment `Regular Expression`: Searches an input text document for a pattern specified in a regular expression (*Regex*) and outputs an XML document containing an XML element for each matching line.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Grep`*: Search text document according to the `pattern` specified in parameter pattern.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.grep.Grep`*: Main class (default)
pattern	Regex pattern being searched for (regular expression) **Possible values:** Any regular expression (e.g. *`.*Test.*`*)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Output

This adapter outputs a UTF-8 encoded XML document with the following structure. For each matching line, an XML element `<Line/>` is created that contains the line number and the content of the respective line:

```
<Text>
   <Line lineNo="Zeilennummer">Inhalt der Zeile</Line>
</Text>
```

10.10.12 Stream Splitter

`Stream Splitter`: Splits a data stream into parts whose length can be defined in byte. This adapter outputs the data stream parts with the specified length successively or outputs the parts in ascending or descending order.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • *`Cut`*: Split data stream and output each part as data stream • *`Clear`*: Reset the counter to the initial value and do not output any part

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.adapter.streamsplitter.StreamSplitter`*: Main class (Default)
BlockSize	Block size in bytes (the last part of the data stream may be smaller) **Possible values:** Any integer number
SegmentCount	Manipulate the adapter result status (useful in combination with a `Condition` component, if only a certain part of the binary data stream, e.g. the last part, has to be processed) **Possible values:** • *`none`*: Do not manipulate the adapter status; Adapter gets the status *`1`*, if a part has been output. (default) • *`increment`*: Output the current counter value as status, beginning with *`1`* • *`decrement`*: Output the current counter value as status, beginning with the highest number and decrement the counter value by *`1`*

Status values

-1(error)	An error occurred during the adapter's execution. You find more detailed information in the server log.

0 (successful)	The adapter was executed successfully, but did not output any data because there were no more parts available (operation `Cut`) or because the adapter did not process the data but only reset the iterator (operation `Clear`).
1 (successful)	The adapter was executed successfully and output the current part of the binary data stream. If the parameter `SegmentCount` has been set to *`increment`*, the adapter has output the first part of the binary data stream.
2 ... n	Only if the parameter `SegmentCount` has been set to *`increment`* or *`decrement`*: The adapter has output the second or the *n*-th part of the binary data stream.

10.10.13 Structured Text

10.10.13.1 Structured Text Converter

`Structured Text Converter`: Converts an XML to a text document or a text to an XML document according to the rules of an ASCII descriptor file.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *`TextToXML`*: Read text document and write it into an XML document • *`XmlToText`*: Write XML data to a text document

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *en.softproject.integration.adapter.text.StructuredTextConverter*: Main class (default)
DescFile	Name of the ASCII descriptor file that describes the text document in its data structure. The ASCII descriptor file should be stored in the user folder of the *X4* repository (e.g. *`X4DB/1/Project/Descr.ad`*). **Possible values:** Path and file name (e.g. *`/Project/Descr.ad`*)

encoding	Character set in which the text document is to be encoded **Possible values:** • Any character string for a character set (e.g. *UTF-8*) • *(empty)*: Use the character set of the processed XML document (default).
Strict	Determines whether the data stream from the preceding process step must be correspondingly long in order to read the complete structure **Possible values:** • `true`: Complete structure must be read • `false`: Data stream can be any length (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.10.13.2 COBOL Copybook Reader

`COBOL Copybook Reader`: Reads a text document in COBOL Copybook format, parses it according to the rules defined in the ASCII descriptor file and outputs an XML document.

> Non-XML characters will be replaced with the *Unicode Replacement Character #xFFFD*. The following Unicode characters are valid XML characters:
>
> • `#x9`
> • `#xA`
> • `#xD`
> • `#x20` – `#xD7FF`
> • `#xE000` – `#xFFFD`

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** `Execute`: Read text document in *COBOL Copybook* format

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.CopyBookReader*: Main class (Default)
DescFile	Name of the ASCII descriptor file specifying the data structure of the COBOL Copybook text document. The ASCII descriptor file should be located in the user directory of the X4 Repository (e.g. *X4DB/1/Project/Descr.ad*) **Possible values:** Path and file name (e.g. */Project/Descr.ad*)
Strict	Defines if the data stream from the preceding process action must be long enough to read the full structure **Possible values:** • *true:* Full structure must be read • *false:* Data stream may have any length (Default)

10.10.13.3 COBOL Copybook Writer

`COBOL Copybook Writer`: Converts an XML document into a text document in *COBOL Copybook* format according to the rules of an ASCII descriptor file.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** *Execute*: Write XML data into a COBOL Copybook text document

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.CopyBookWriter*: Main class (Default)
DescFile	Name of the ASCII descriptor file specifying the data structure of the *COBOL Copybook* text document. The ASCII descriptor file should be located in the user directory of the X4 repository (e.g. *X4DB/1/Project/Descr.ad*) **Possible values:** Path and file name (e.g. */Project/Descr.ad*)

encoding	Character encoding of the *Cobol Copybook* text document to be created **Possible values:** • Any valid character encoding string (e.g. *UTF-8*) • *(empty)*: Use the input XML document's character encoding (Default)
DebugMode	Debug mode

10.10.14 Text Replacement

? Unknown Attachment `Text Replacement`: Searches a text document for a character string (case-sensitive) and outputs a text document with the same character encoding in which each searched character string has been replaced by a new character string.

> If you want to replace a character string in an XML document, note that the adapter always outputs a plain text document, so it is no longer necessarily well-formed XML!
> The character encoding of the document must be set explicitly, e.g. via the *Document Properties Changer*.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *Replace*: Replace string

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.replace.Replace*: Main class (default)
searchString	String to search for in the input text document **Possible values:** Any string
replaceString	String by which `searchString` is to be replaced in the result text document. **Possible values:** Any string

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

This adapter expects any text document.

Output

The adapter outputs a text document in which the found text was replaced.

10.10.15 Transliteration Adapter

`Transliteration Adapter`: Transliterates text from one format to another, e.g. from UTF-8 to LATIN1 characters (ä → ae etc.). However, it doesn't translate texts into other languages, nor changes the charset.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `GetAvailableIDs`: Lists all available transliterators • `Transliterate`: Transliterates the text according to a specified transliterator (id) • `TransliterateFromRule`: Transliterates the text according to a given rule string (see http://userguide.icu-project.org/transforms/general/rules)

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter. transliteration.TransliterationAdapter*: Main class (default)
`id`	Id of the used transliterator **Possible values:** A value delivered by the *`GetAvailableIDs`* operation
`rule`	Rule used to transliterate **Possible values:** One of the rules specified under http://userguide.icu-project.org/transforms/general/rules

Status values

`1`	The adapter was executed successfully
`-1`	An error occurred during the adapter's execution

Input

The adapter expects a text or XML document as input.

Output

The adapter outputs the transliterated text or XML document.

10.11 Geo Location

- Bing Maps
- Geo Calculation
- Google Maps

10.11.1 Bing Maps

`Bing Maps`: Accesses *Bing Maps* via its web service interface in version 1 and retrieves the geographical information for an address. In addition, a route including waypoints can be calculated and a URL to the corresponding *Bing Maps* map section can be generated. The adapter outputs its results as XML or JSON document.

> In order to use the *Bing Maps API* a *Bing Maps API Key* is required.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *`Geocode`*: Retrieve all geographical information for an address (street, country/state, nation, coordinates etc.) • *`ReverseGeocode`*: Retrieve all geographical information for a decimal coordinate (street, country/state, nation, coordinates etc.) • *`GetDirections`*: Retrieve the route between a start point and a destination with any number of waypoints • *`CreateGoogleURL`*: Create a URL to a Bing Maps map section image

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.geo.microsoft.BingMapsAdapter*: Main class (Default)
bingMapsKey	Key for using the *Bing Maps API*

outputFormat	Define the output format for requests **Possible values:** • `xml`: Output XML document • `json`: Output JSON document
timeout	Timeout to establish a connection in milliseconds **Possible values:** • Any integer number • `0`: No limitation • `20000`: Cancel connection establishment after 20 seconds
pxHost	IP address or host name of a proxy server **Possible values:** Any valid string (e. g. `192.168.1.1`)
pxPort	Port number of the proxy server **Possible values:** Any integer number
pxLogin	User name for the proxy server **Possible values:** Any valid string
pxPwd	Password for the proxy server **Possible values:** Any valid string

Status values

1 (successful)	The request was successful and a result was output (corresponds to the HTTP status code 200)
0 (empty)	The request was successful, but no result was output
-1 (failed)	The request failed because of a technical error
- HTTP status code	The request was executed, but terminated with a status code unequal 200. Error interpretation corresponds to the HTTP status code.

For more information on operations, see:

- Geocode
- ReverseGeocode
- GetDirections
- CreateBingURL

10.11.1.1 Operation: Geocode

`Bing Maps` **adapter**

The operation `Geocode` allows to retrieve all geographical information for an address.**Input**

For the operation `Geocode` the adapter expects an input XML document containing a complete address (for incomplete addresses the request may not be unambiguous) and having the following structure:

```
<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>
```

Output

For the operation `Geocode` an XML or JSON document will be output that contains the geographical information for the requested address.

10.11.1.2 Operation: ReverseGeocode

`Bing Maps` **adapter**

The operation `ReverseGeocode` allows to retrieve all geographical information for a decimal coordinate.**Input**

For the operation `ReverseGeocode` the adapter expects an input XML document with a decimal coordinate and having the following pattern:

```
<ReverseGeocode>
    <LatLng>
        <data><!--Decimal coordinate in format NN.NNNNNNNNNNNNNN,
EE.EEEEEEEEEEEEEE--></data>
    </LatLng>
</ReverseGeocode>
```

Output

For the operation `ReverseGeocode` an XML or JSON document will be output that contains the geographical information for the requested geodecimal coordinate.

10.11.1.3 Operation: GetDirections

`Bing Maps` **adapter**

The operation `GetDirections` allows to calculate the route between a start point and a destination with any number of waypoints.**Input**

For the operation `GetDirections` the adapter expects an input XML document which contains start, destination and any number of waypoints as decimal coordinate or address and having the following structure:

```
<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>
```

Output

For the operation `GetDirections` an XML or JSON document will be output which contains a route for cars. If more than one route is available, the fastest route will be output.

10.11.1.4 Operation: CreateBingURL

`Bing Maps` **adapter**

With the operation `CreateBingURL` a URL to a *Bing Maps* map section image can be created.**Input**

For Operation `CreateBingURL`, the adapter expects an XML input document that defines the URL part of the *Bing Static Maps API* in the `Path` element and the map section to be created in various `Param` elements. A parameter with attribute `name="mapSize"` contains the pixel dimensions of the map section to be created.

Sample 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>
```

For more information on creating and formatting *Bing Maps* map cutout graphics, see Bing Static Map API at https://msdn.microsoft.com/en-us/library/ff701724.aspx.

Output

For the operation `CreateBingURL` an XML document will be output that contains a URL for a *Bing Maps* map section that corresponds to the input. The `outputFormat` is passed to the `o` parameter in the URL.

```
<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

? Unknown Attachment `Geo Calculation`: Converts polylines or polygons, whose waypoints are defined in the formats *GeoDecimal, GeoE5, GooglePoint, GoogleLine, PtvLine* or *PtvPoint* into the formats *GeoDecimal, GeoE5, GooglePoint* or *GoogleLine,* it calculates the line (route) length and edge length of polygones, (edge) intersections between polylines/polygons for calcutions of territory boundaries, and it calculates bounding boxes for one or multiple polygons/polylines.
For a line that intersects a polygon, the line segments inside and outside of the polygon can be calculated, e. g. to differentiate between trips inside and outside of a territory. In addition, polylines or polygons can be simplified in order to remove some waypoints that are not necessary for a map view.

Limitations
When calculating polygons that span more than an earth hemisphere (more than 180°), unexpected results may occur. Line segments cannot span more than 170°. If you need to perform such operations, contact the SoftProject support via support@softproject.de.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • *`LineConversion`*: Convert polylines or polygons, whose waypoints are defined in the formats *GeoDecimal, GeoE5, GooglePoint, GoogleLine, PtvLine* or *PtvPoint* into the formats *GeoDecimal, GeoE5, GooglePoint* or *GoogleLine* • *`LineLength`*: Calculate the line (route) length and edge length of polygones in kilometers • *`LineIntersection`*: Calculate the (edge) intersections between polygons or polylines • *`BoundingBox`*: Calculate a bounding box (smallest possible rectangle according to the mercator cylinder projection in the plane) for polygons or polylines, and output each with the latitude/longitude of the corners • *`SegmentLineByPolygon`*: Calculate the line segments inside and outside of a polygon for a line that is intersected by a polyline • *`LineSimplificationPercent`*: Simplify polylines/polygons proportionally to the total route in order to remove waypoints not necessary for a map view • *`LineSimplificationKilometer`*: Simplify polylines/polygons in order to remove waypoints that are not necessary for a map view

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.adapter.geo.calc.GeoCalculationsAdapter`*: Main class (Default)

outputFormat	The calculated geodata output format **Possible values:** • `GeoDecimal`: Geodecimally-encoded coordinates with latitude/longitude (e.g. `48.952/8.391`) • `GeoE5`: Geo E5-encoded coordinates with latitude/longitude (e.g `4895176/839096`) • `GooglePoint`: *GooglePoint-encoded coordinates (e.g.* `o{wiHozer@`) • `GoogleLine`: *GoogleLine*-encoded polyline consisting of GooglePoint waypoints (e.g. `o{wiHozer@or@ow@_yF?`)
percent	Only for operation `LineSimplificationPercent`: width of a corridor that is defined relatively to the total route length in percent divided by 100. Within this corridor, the polyline/polygon will be simplified. **Possible values:** • Any number between 0 and 1 with a dot as decimal separator (e.g. `0.5`) • `0.01`: Remove all additional waypoints within a corridor that has a width equaling 1 percent (default)
kilometer	Only for operation `LineSimplificationKilometer`: width of a corridor in kilometers. Within this corridor, the polyline/polygon will be simplified. **Possible values:** • Any positive integer number or 0 with a dot as decimal separator (e.g. `0.5`) • `0.01`: Remove all additional waypoints within a corridor that has a width of 10 meters (default)

Status values

1 (successful)	The adapter operation has been executed successfully
0 (empty)	For operation `LineIntersection`: the number of polylines is not 2; For operation `SegmentLineByPolygon`: the input does not contains any element Polygon, Polyline or both elements are missing, or an geometrical error has occurred (polygon with less than tree points, polyline with less than tow points)

-1 (failed)	The adapter input is no well-formed XML document

Input

This adapter expects consistently specified geodata for points and lines within the input XML document. Geodecimally-encoded coordinates must be specified as follows:

```
<GeoDecimal>
   <Latitude><!--Latitude with the format NN.NNNNN--></Latitude>
   <Longitude><!--Longitute with the format EE.EEEEE--></Longitude>
</GeoDecimal>
```

Geo E5-encoded coordinates must be specified as integer values (geodecimally-encoded coordinate value multiplied by 100.000):

```
<GeoE5>
   <Latitude><!--Latitude with the format NNNNNNN--></Latitude>
   <Longitude><!--Longitute with the format EEEEEEE--></Longitude>
</GeoE5>
```

Google Point-encoded coordinates must be specified as follows:

```
<GooglePoint><!--GooglePoint code--></GooglePoint>
```

Google Line-encoded polylines must be specified as follows:

```
<GoogleLine><!--GoogleLine code--></GoogleLine>
```

PTV Point-encoded coordinates (in PTV Mercator projection) must be specified as follow:

```
<PtvPoint>
   <X><!--Easting as number with optional comma separator--></X>
   <Y><!--Northing as number with optional comma separator--></Y>
</PtvPoint>
```

PTV Line-encoded polylines (in PTV Mercator projection) must be specified as follows. 1-n value pairs consisting of easting and northing are concatenated so that an even number of values is deposited. Each value must be separated by a comma.

```
<PtvLine><!--Easting1,Northing1,Easting..,Northing...--></PtvLine>
```

10.11.2.1 Operation: Line Conversion

With the operation `LineConversion` polylines or polygons whose waypoints have been specified in the formats `GeoDecimal`, `GeoE5`, `GooglePoint`, `GoogleLine`, `PtvLine` or `PtvPoint` can be converted into the formats `GeoDecimal`, `GeoE5`, `GooglePoint` or `GoogleLine`.

Input

For the operation `LineConversion` the adapter expects an input XML structure with the following pattern that may contain any number of polygons (each within an element `Polygon`) or polylines (each within an element `Polyline`) and whose corner/waypoints are specified in the formats `GeoDecimal`, `GeoE5`, `GooglePoint`, `GoogleLine`, `PtvLine` or `PtvPoint` within a child element of the same name.

For each polygon or polyline you can define any attribute that will be output in the result XML document without modification.

```
<AnyRoot>
   <!-- 0-n polylines-->
   <Polyline attribute="value"> <!-- Any attributes -->
       <!--1-n point definitions in the formats GeoDecimal, GeoE5, GooglePoint or
PtvPoint-->
   </Polyline>

   <!-- 0-n polygones -->
   <Polygon attribute="value"> <!-- Any attributes -->
       <!--1-n point definitions in the formats GeoDecimal, GeoE5, GooglePoint or
PtvPoint-->
   </Polygon>
</AnyRoot>
```

Output

For the operation `LineConversion` an XML document will be output that has the same structure as the input document and each attribute that has been defined within the elements `Polygon` or `Polyline` will be output without modification. Depending on the selected output format, the point definitions in the corresponding elements are converted into the format `GeoDecimal`, `GeoE5` or `GooglePoint`; or polylines are converted into the `GoogleLine` format.

10.11.2.2 Operation: LineLength

With the operation `LineLength` the length of polylines or the edge length of polygons can be calculated in kilometers.

Input

For the operation `LineLength` the adapter expects the same input XML structure as for the operation `LineConversion`, where polygones and/or polylines are specified as well as additional (optional) attributes.

Output

For the operation `LineLength` an XML document with the following structure will be output. For each element `Polygon` or `Polyline` that has been defined within the input XML structure and that contains a polygon or polyline, an element `Length` will be generated that specifies the polyline length or the polyline edge length in kilometers.

All attributes that were defined in the input XML within the elements `Polygon` or `Polyline` will be output without modification.

```
<Data>
   <!-- For each polygon/polyline element -->
   <Length attribut="value"><!--Length in kilometers (with 3 decimal places)--></
Length>
</Data>
```

10.11.2.3 Operation: LineIntersection

With the operation `LineIntersection` the (edge) intersections between polygons or polylines can be calculated.

Input

For the operation `LineIntersection` the adapter expects the same input XML structure as for the operation `LineConversion`, where two polygones, two polylines or a polygon and a polyline are specified. Additional attributes are not taken over to the output XML document.

Output

For the operation `LineIntersection` an XML document will be output that contains all calculated (edge) intersections within the element `Intersection.` Depending on the selected output format, the intersection points will be output in a child element `GeoDecimal`, `GeoE5` or `GooglePoint`.

As an alternative, with the output format `GoogleLine` a line that leads along the intersection points can be output within a child element of the same name:

```
<Data>
   <Intersection>
      <!-- 0-n Intersection points/lines -->
      <Point/line format><!--Point/line definition--></Point/line format>
   </Intersection>
</Data>
```

10.11.2.4 Operation: BoundingBox

With the operation `BoundingBox` a bounding box (the smallest possible rectangle according to the mercator cylinder projection in the plain) for polygons or polylines can be calculated, and the corner's longitude/latitude can be output.

Input

For the operation `BoundingBox` the adapter expects the same input XML structure as for the operation `LineConversion`, where polygones, polylines as well as additional (optional) attributes are specified.

Output

For the operation `BoundingBox` an XML document will be output, where for each element `Polygon` or `Polyline` that has been defined within the input XML the corners of a rectangular bounding box will be output. In addition, all attributes that were defined in the input XML within the elements `Polygon` or `Polyline` will be output without modification.

The required corners (latitude/longitude) to spread out a bounding box will be output as follows:

```
<Data>
   <!-- For each polygon/polyline element -->
   <BoundingBox attribute="value"> <!-- Any attributes -->
      <MinLatitude><!--Geodecimally-encoded latitude of corner 1--></MinLatitude>
      <MaxLatitude><!--Geodecimally-encoded latitude of corner 2--></MaxLatitude>
      <MinLongitude><!--Geodecimally-encoded longitude of corner 1--></MinLongitude>
      <MaxLongitude><!--Geodecimally-encoded longitude of corner 2--></MaxLongitude>
   </BoundingBox>
</Data>
```

10.11.2.5 Operation: SegmentLineByPolygon

With the operation `SegmentLineByPolygon` the line segments inside or outside of a polygon that is intersected by a polyline can be calculated.

Input

For the operation `SegmentLineByPolygon` the adapter expects the same input XML structure as for the operation `LineConversion`, where you define one polygon within the element `Polygon` and one polyline within the element `Polyline`. Additional attributes will not be taken over to the output XML document.

Output

For the operation `SegmentLineByPolygon` an XML document will be output, where the elements `OriginalPolygon` and `OriginalPolyline` contain the points of the original polygon and the polyline in the selected output format, as well as its (edge) length in kilometers within the attribute `length`.

Within the element `FinalPolygon` and `FinalPolyline` the corrected points of the polygon and the polyline will be output in the selected output format. In addition, the polygon's and polyline's (edge) length will be output in kilometers within the attribute `length`.

Within the element `Inside` for each line segment that is located inside a polygon, an element `Part` will be output that contains all points. Depending on the selected output format, the points will be output in a child element `GeoDecimal`, `GeoE5` or `GooglePoint`. In addition, within each attribute `length`, the length of the line segments inside will be output in kilometers.

Within the element `Outside`, for each line segment that is located outside of the polygon an element `Part` will be output that contains all points.Depending on the selected output format, the points will be output in a child element `GeoDecimal`, `GeoE5`, `GooglePoint` or as `GoogleLine`. In addition, within each attribute `length`, the length of the line segments outside will be output in kilometers.

Within the element `Edge`, lines will be output that are located exactly on the intersection line.

```
<Result>
   <OriginalPolyline length="Length of the original polyline in kilometers">
      <!-- 0-n points/lines -->
      <Point/Line format><!--Point/Line definition--></Point/Line format>
   </OriginalPolyline>
   <OriginalPolygon length="Edge length of the original polygon in kilometers">
      <!-- 0-n points/lines -->
      <Point/Line format><!--Point/Line definition--></Point/Line format>
   </OriginalPolygon>
   <FinalPolyline length="Length of the corrected polyline in kilometers">
      <!-- 0-n points/lines -->
      <Point/Line format><!--Point/Line definition--></Point/Line format>
   </FinalPolyline>
   <FinalPolygon length="Edge length of the corrected polygons in kilometers">
      <!-- 0-n points/lines -->
      <Point/Line format><!--Point/Line definition--></Point/Line format>
   </FinalPolygon>
   <Inside>
      <!-- 0-n line segments-->
      <Part length="Length of the inside polyline in kilometers">
         <!-- 1-n points/lines -->
         <Point/Line format><!--Point/Line definition--></Point/Line format>
      </Part>
   </Inside>
   <Outside>
      <!-- 0-n line segments-->
      <Part length="Length of the outside polyline in kilometers">
         <!-- 1-n points/lines -->
         <Point/Line format><!--Point/Line definition--></Point/Line format>
      </Part>
   </Outside>
   <Edge>
      <!-- 0-n line segments-->
      <Part length="Length of the line segment lying exactly on the line in
kilometers">
         <!-- 1-n points/lines -->
         <Point/Line format><!--Point/Line definition--></Point/Line format>
      </Part>
   </Edge>
</Result>
```

10.11.2.6 Operation: LineSimplificationPercent / LineSimplificationKilometer

With the operation `LineSimplificationPercent` and `LineSimplificationKilometer` polylines/ polygons can be simplified according to the Douglas Peucker algorithm in order to remove waypoints within a corridor that are not necessary for a map view. The corridor width can be defined relative to the total route or as absolute value in kilometers.

Input

For the operation `LineSimplificationPercent` and `LineSimplificationKilometer` the adapter expects the same input XML structure as for operation `LineConversion`, where you can define any

polylines, each within an element `Polyline`, or any polygones, each within an element `Polygon`. Additional attributes will be taken over to the result XML document.

Output

For the operation `LineSimplificationPercent` and `LineSimplificationKilometer` an XML document will be output that has the same structure as the input XML document and contains all attributes that have been defined within the element `Polygon` or `Polyline` without modification. Depending on the corridor width (defined within the parameters `percent` or `kilometer`), the polylines/polygons will be simplified. Depending on the selected output format, the point definitions will be converted into the formats `GeoDecimal`, `GeoE5` or `GooglePoint`; or polylines will be converted into the `GoogleLine` format and deposited in the corresponding elements.

10.11.3 Google Maps

? Unknown Attachment `Google Maps`: Accesses *Google Maps* via its web service interface and retrieves the geographical information for an address. In addition, a route including waypoints can be calculated and a URL to the corresponding Google Maps map section can be generated. The adapter outputs its results as XML or JSON document.

> By default, the free *Google Maps API* is used; To benefit from all services without limitations, you can also use the *Google Maps API Premier* with this adapter by specifying credentials within the parameters `clientID` and `privateKey`.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • *`Geocode`*: Retrieve all geographical information for an address (street, country/state, nation, coordinates etc.) • *`ReverseGeocode`*: Retrieve all geographical information for a decimal coordinate (street, country/state, nation, coordinates etc.) • *`GetDirections`*: Retrieve the route between a start point and a destination with any number of waypoints • *`CreateGoogleURL`*: Create a URL to a Google Maps map section image (when using the *Google Maps API Premier* with a signed URL)

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.geo.google.Google MapsAdapter*: Main class (default)
clientID	Client ID to use the *Google Maps API Premier* **Possible values:** • Any valid string • `(empty)`: Use free Google Maps API (default)
privateKey	Private key for the *Google Maps API Premier* (if parameter `clientID` is set) **Possible values:** Any valid string
outputFormat	Define the *Google Maps* output format for requests **Possible values:** • *xml*: Output XML document • *json*: Output JSON document
timeout	Timeout to establish a connection in milliseconds **Possible values:** • Any integer number • *0*: No limitation • *20000*: Cancel connection establishment after 20 seconds
pxHost	IP address or host name of a proxy server **Possible values:** Any valid string (e.g. *192.168.1.1*)
pxPort	Port number of the proxy server **Possible values:** Any integer number
pxLogin	User name for the proxy server **Possible values:** Any valid string

pxPwd	Password for the proxy server **Possible values:** Any valid string

Status values

1 (successful)	The request has been successful and a result has been output
0 (empty)	The request has been successful, but no result has been output
-1 (failed)	The request failed because of a technical error

For more information on operations, see:

- Geocode
- ReverseGeocode
- GetDirections
- CreateGoogleURL

10.11.3.1 Operation: Geocode

`Google Maps` **adapter**

With the operation `Geocode` all geographical information for an address (street, country/state, nation, coordinates etc.) can be retrieved.

Input

For the operation `Geocode` the adapter expects an input XML structure with the following pattern containing a complete address (for incomplete addresses the request may not be unambiguous):

```
<Geocode>
   <Address>
      <Street><!--Street and house number--></Street>
      <ZIP><!--ZIP code--></ZIP>
      <City><!--Place--></City>
      <Country><!--Country--></Country>
   </Address>
</Geocode>
```

Output

For the operation `Geocode` an XML or JSON document will be output that contains the geographical information (street, country/state, nation, coordinates etc.) for the requested address according to http://code.google.com/intl/en/apis/maps/documentation/geocoding/#XML.

10.11.3.2 Operation: ReverseGeocode

`Google Maps` **adapter**

With the operation `ReverseGeocode` all geographical information for a decimal coordinate (street, country/state, nation, coordinates etc.) can be retrieved.

Input

For the operation `ReverseGeocode` the adapter expects an input XML structure with the following pattern that contains a decimal coordinate:

```
<ReverseGeocode>
   <LatLng>
      <data><!--Decimal coordinate in format NN.NNNNNNN, EE.EEEEEEE--></data>
   </LatLng>
</ReverseGeocode>
```

Output

For the operation `ReverseGeocode` an XML or JSON document will be output that contains the geographical information (street, country/state, nation, coordinates etc.) for the requested geodecimal coordinate according to http://code.google.com/intl/en/apis/maps/documentation/geocoding/#XML.

10.11.3.3 Operation: GetDirections

`Google Maps` **adapter**

With the operation `GetDirections` the route between a start point and a destination with any number of waypoints can be calculated.

Input

For the operation `GetDirections` the adapter expects an input XML structure with the following pattern that contains start, destination and any number of waypoints as decimal coordinate, address or in the GooglePoint format:

```
<GoogleRouting xmlns:exslt="http://exslt.org/common">
 <Origin><!--Decimal coordinate, address/POI or GooglePoint--></Origin>
 <Waypoints>
    <!-- Any number of waypoints -->
    <Waypoint><!--Decimal coordinate, address/POI or GooglePoint--></Waypoint>
 </Waypoints>
 <Destination><!--Decimal coordinate, address/POI or GooglePoint--></Destination>
</GoogleRouting>
```

Addresses are specified in the following XML structure:

```
<Address>
  <Street><!--Street and house number--></Street>
  <ZIP><!--ZIP code--></ZIP>
  <City><!--Place--></City>
  <Country><!--Country--></Country>
</Address>
```

Locations (points of interest, etc.) are specified in the following XML structure:

```
<Address>
   <Location><!--Point of Interest--></Location>
   <City><!--Place</City>
</Address>
```

GooglePoints specified in the following XML element:

```
<GooglePoint><!--GooglePoint code--></GooglePoint>
```

Decimal coordinates are specified in the following XML structure:

```
<GeoDecimal>
   <Latitude><!--NN.NNNNNNN--></Latitude>
   <Longitude><!--EE.EEEEEEE--></Longitude>
</GeoDecimal>
```

Output

For the operation `GetDirections` an XML or JSON document will be output that contains a route for cars according to http://code.google.com/intl/en/apis/maps/documentation/directions/#XML. If more than one route is available, the fastest route will be output.

10.11.3.4 Operation: CreateGoogleURL

`Google Maps` **adapter**

With the operation `CreateGoogleURL` a URL to a *Google Maps* map section image (when using the *Google Maps API Premier* with a signed URL) can be created.

Input

For the operation `CreateGoogleURL` the adapter expects an input XML structure with the following pattern that contains the URL part of the *Google Static Maps API* within the element `Path` (see http://code.google.com/intl/de-DE/apis/maps/documentation/staticmaps/). In addition, the map section to be displayed is specified within `Param` elements. One of these parameters has an attribute `name="size"` containing the pixel measurements of the map section to be displayed.

Each `Param` element with the attribute `name="markers"` defines a marker with a color (e. g. `color:blue`) and a text label (a capital letter, e. g. `label:X`).

A `Param` element with the attribute `name="path"` defines path properties (e. g. line color `color:blue`) separated by |, as well as a route having the prefix `enc:` that consists of *GoogleLine*-encoded waypoints:

```
<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>
```

Output

For the operation `CreateGoogleURL` an XML document will be output that contains a URL for a *Google Maps* map section that corresponds to the input. The `outputFormat` has no effect on the composition of the 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`: Reads an *Audatex DV90* calculation document and outputs the data as XML document.

Properties

Operation	Defines the operation executed by this native adapter **Possible values:** *`ConvertToXml`*: Read *Audatex DV90* text document

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.nativeadapter.NativeAdapter`*: Main class (Default)

Statuswerte

`1 (`*`successful`*`)`	The *DV90* calculation document was successfully converted.
`-1 (`*`failed`*`)`	Conversion has been failed due to a technical error.

10.12.2 GDV

10.12.2.1 GDV Message Plausibility Checker

`GDV Message Plausibility Checker`: Checks a GDV message for plausibility according to the rules of a configuration file

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Execute`*: Check GDV message for plausibility, according to the rules of a configuration file specified in parameter `iniFile`.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.gdvplausi.GDVPlausiPrueferWrapper*: Main class (default)
iniFile	Path to the configuration file (`.ini` format) **Possible values:** path and file name (e.g. */Project/GDV.ini*)

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

10.12.2.2 GDV Message Reader

`GDV Message Reader`: Reads a GDV message, parses it according to the rules of a descriptor file and outputs an XML document with UTF-8 character encoding. Supported message types are SN 2000 / 2003, VU/VM 2009 / 2007 / 2005 / 2002 / 1998.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *Execute*: Read GDV message and convert it to XML, according to the rules of a descriptor file specified in parameter `descriptorFile`.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.ASCIIGDV*: Main class (default)
DescFile	Descriptor file describing the GDV message in its data structure, see *X4 Designer manual - ASCII Descriptor*. The descriptor file should be stored in the user folder of the X4 repository (e.g. *X4DB/1/Project/Descr.ad*) **Possible values:** Path and file name (e.g. */Project/Descr.ad*)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.12.2.3 GDV Message Writer

`GDV Message Writer`: Converts an XML document to a GDV message according to the rules of a descriptor file.

Properties

Operation	Determines which operation the adapter executes **Mögliche Werte:** *`Execute`*: Generate GDV message from an XML document, according to the rules of a descriptor file specified in parameter `DescFile`.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.ASCIIGDV`*: Main class (default)
DescFile	Name of the descriptor file that describes the GDV message in its data structure, *see X4 Designer manual - ASCII descriptor*. The descriptor file should be stored in the user folder of the X4 repository (e.g. *`X4DB/1/Project/Descr.ad`*). **Possible values:** Path and file name (e.g. */Project/Descr.ad*)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.12.2.4 GDV-VU

`GDV-VU`: The GDV VU Adapter can be used to read and convert data in GDV text format.

Two options are available.

- Conversion of files in GDV ASCII format to XML files
- Conversion of XML files to files in GDV ASCII format

In addition, the adapter can be used to adjust the date fields and the decimal number fields in the GDV file.

Date fields

GDV date fields can be 8, 6, 4 or 2 characters long and are converted to an ISO date format by the adapter as follows

	GDV format	ISO format	Example		Notice
8	12102022	TT-MM-JJJJ	12102022	12-01-2022	Even incomplete dates like 00002008 are converted to the appropriate format 00-00-2008
6	092022	MM-JJJJ	092022	09-2022	
4	1022	MM-JJ	1022	10-22022	Even incomplete dates like 0000 are converted to the corresponding format 00-00
2	14	TT	14	14	

⚠ When converting XML to GDV format, the data in ISO format is adapted to the length of the GDV field. If the date in ISO format is longer than the GDV field, the date will be shortened. For example, if an ISO date 2022-03-24 is converted to a GDV field with a length of 4 characters, a date 0322 is inserted and the days of the original format are ignored.

Decimal number fields

GDV decimal number fields consist of the length of the total number and the number of decimal places. This means, for example, that when converted to XML format, a number field with length "4" and two decimal places becomes 54.34.

Number of decimal places	GDV content	Converted value
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

⚠ When converting XML to GDV format, the numbers are adapted to the length of the GDV field. If a field in XML format is shorter than the length defined for the GDV field, the numbers are supplemented with preceding blanks.

Properties

`Operation`	Determines which operation the adapter executes **Possible values:** • *`GDV to XML`*: Converts the GDV data into an XML format. • *`XML to GDV`*: Converts the XML data into a GDV format. • *`Get supported sets`*: Displays all available "sets" of the selected GDV format.

Parameter

`formattedXmlValues`	Defines the formatting of the date and number fields during import/export **Possible values:** • `Active`: All GDV numbers and date fields are adapted as described. GDV_Date will be converted to an ISO date. GDV-NumField is converted to an integer or decimal number as defined in the data type. • `Inactive`: Data in the target document is taken over as in the source document, which means that the data in the XML document must already be in GDV format.
`trimXmlValues`	Defines whether spaces are deleted before or after the output value. Which spaces are deleted depends on the alignment. This does not apply when an XML file is imported. **Possible values:** • `Active`: Blank spaces in the output values are deleted. The deletion of the blanks depends on the set alignment - this means that only those blanks are deleted which are on the undefined side (unaligned). • `Inactive`: The blank spaces are left in the output file.
`createHeaderFooter`	Defines whether a header and footer are created in the output file with the values `<Vorsatz>` and `<Nachsatz>`. **Possible values:** • `Active`: Vorsatz and Nachsatz are created in the XML or GDV file. • `Inactive`: The elements Vorsatz and Nachsatz are not created in the XML or GDV file.

gdvVersion	Defines with which GDV version the data will be imported/exported. Displays the selected GDV version **Possible values:** • 2009 – GDV-Version 2009 • 2013 – GDV-Version 2013 • 2015 – GDV-Version 2015 • 2018 – GDV-Version 2018

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Output

Depending on the selected adapter operation, an XML document or a GDV document is output.

The `XML to GDV` operation generates a GDV file if the input data is correct.

If the input data is incorrect, an XML file with errors is output.

10.12.3 SZF Reader

? Unknown Attachment `SZF Reader`: Reads a *SilverDAT II* calculation document and outputs the data as an XML document.
Naming of the adapter up to version 5.5.4: *SZF Converter*

Properties

Operation	Determines which operation the adapter executes **Possible values:** *Read*: Read *SilverDAT II* text document

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.szfreader.SzfReader*: Main class (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

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`: Connects to an MMS server and reads the device and node information.

The connector thus provides a basic interface for the internationally established communication standard IEC 61850 for the automation of switchgears. It acts as a client and can receive messages as classic client-server communication.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • *`RequestServerNodes`*: Search for certain devices and/or nodes and read the values • *`RequestServerModel`*: Query the device and node directory for the MMS server and output it as XML

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.iec.IEC61850* : Main class (default)
address	The MMS server's IP address or host name **Possible values:** Any valid string (e. g. *`192.168.1.1`*)
port	The MMS server's port number **Possible values:** Any integer
authenticationParameter	Optional login parameter to log onto the MMS server

responseTimeout	Time limit for the server request in milliseconds **Possible values:** • Any integer • *0*: No time limit • *20000*: Cancel server request after 20 seconds
messageFragmentTimeout	Time limit between two incoming data packages in milliseconds **Possible values:** • Any integer • *0*: No time limit • *20000*: Cancel connection after 20 seconds
maxTPDU	Sets the maximum TPDU (Transfer Protocol Data Unit) **Possible values:** Any integer
maxMMSPDU	Sets the maximum MMS PDU **Possible values:** Any integer between 64 and 65000

Status values

1 (successful)	The request was successful and a result was returned
-1 (failed)	The request failed due to a technical error

Input

Operation RequestServerModel

The operation `RequestServerModel` doesn't require any input XML document.

Operation RequestServerNodes

The operation `RequestServerNodes` expects an input XML document, which is structured as follows:

```
<RootElement>
    <!-- @Fc is optional; @ObjectReference is required -->
    <Node ObjectReference="ied1lDevice1/LPHD1"/>        <!-- Returns all nodes and
devices within ied1lDevice1/LPHD1 -->
    <Node ObjectReference="ied1lDevice1/LPHD1" Fc="MX"/><!-- Returns all nodes and
devices within ied1lDevice1/LPHD1 with the function code MX  -->
</RootElement>
```

Output

Operation RequestServerModel

The operation `RequestServerModel` returns an XML document, which is structured as follows:

```
<?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

The operation `RequestServerNodes` returns an XML document, in which the corresponding node of the input XML is extended by the response:

```
<?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`: Connects to devices via the Modbus TCP protocol.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Read`: Reads data from the Modbus server • `Write`: Writes data to the Modbus server

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.modbus.ModBusConnectorAdapter*: Main class (default)
`ip`	Server IP
`port`	Server port Possible values: • any port number • *502* (default)

Status values

`1`	The operation was executed successfully
`-1`	Any error occurred during the operation's execution

Input

The adapter expects a specific XML structure for the different operations:

- *Operation `Read`:*

Sample input for 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`:*

Sample input for 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>
```

Output

The adapter outputs different structures depending on the executed operation:

- *Operation `Read`:*

Sample output for 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`:*

Sample output for 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>
```

Sample output in case of an error

```
<?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>
```

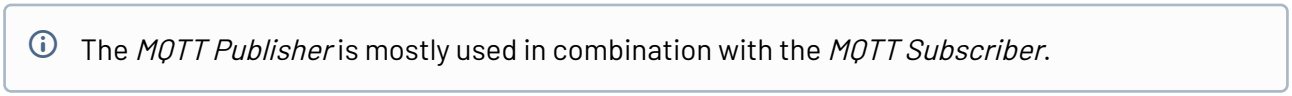
10.13.3 MQTT Publisher/Subscriber

- MQTT Publisher
- MQTT Subscriber

10.13.3.1 MQTT Publisher

`MQTT Publisher`: Allows to publish messages in a MQTT broker.

The MQTT protocol provides a simple method for sending messages via a publish/subscribe model. This makes it particularly suitable for messaging in the internet of things environment, such as low-power sensors or mobile devices like telephones, embedded computers or micro-controllers.



Properties

`Operation`	Defines the operation executed by the adapter. **Possible values:** • `Publish`: Publish a message in a MQTT broker

Parameters

`Adapter`	Adapter main class (do not change!) **Possible values:** *de.softproject.integration.adapter.mqtt.MqttPublisherAdapter*: Main class (default)
`Alias`	Broker alias **Possible values:** Any string ⓘ The alias can be defined within the configuration file `mqttconfig.xml` as attribute of the element `Broker`.
`ClientId`	Identifier for the client that publishes in the broker **Possible values:** • If the Client ID is defined in the configuration file `mqttconfig.xml`: Any string • If the Client ID is not defined in the configuration file `mqttconfig.xml`: UUID ⓘ The Client ID can be defined within the configuration file `mqttconfig.xml` for each broker in the `ClientId` element.
`clientTimeout`	Timeout for ONLINE_ON_DEMAND connections (for the operation `Publish`). If `ConnectionMode` differs from `ONLINE_ON_DEMAND`, `ClientTimeout` is not taken into account. **Possible values:** Integer greater than `0`
`connectionMode`	Connection mode selected to create the Client. Depending on the selected value, the client connection is held or disconnected after a certain time (`ClientTimeout`). **Possible values:** • `ONLINE_ON_DEMAND`: The `ClientTimeout` parameter is required to configure the client. • `ALWAYS_ONLINE_ON_FIRST_PUBLISH`: The client stays connected as soon as an initial publishing happens. Therefore, the `ClientTimeout` parameter is not required. ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.

`stateful`	Defines if the state between client and broker is stateful. **Possible values:** `true` / `false` ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.
`keystorePath`	Path to the keystore. If no SSL/TLS configuration is required to connect to the broker, this parameter value may be empty. **Possible values:** File system path ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.
`keystorePassword`	Password for the keystore specified for `KeystorePath`. **Possible values**: Any string ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.
`Topic`	Name of the topic in which the client publishes. **Possible values:** Any string ⓘ The topic can be defined within the configuration file `mqttconfig.xml` for each broker in the attribute `filter` of the element `Topic`.
`server`	Protocol, host and port to connect client and broker. **Possible values:** • `tcp://hostname:port`: If no SSL/TLS connection is used. • `ssl://hostname:port`: If a SSL/TLS connection is used. ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.

`InputFormat`	Input format of the adapter **Possible values:** • `BYTES` • `BASE64_STRING` • `BASE64_XML`
`loginUsername`	Username that is used to connect to a broker that requires credentials. **Possible values:** Username ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.
`loginPassword`	Password that is used to connect to a broker that requires credentials. **Possible values:** Password ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.
`retained`	Defines whether the published message for the specific topic is kept and stored in the broker. Any client subscribing to this topic will receive the kept message immediately after subscribing. **Possible values:** `true` / `false`

Status values

`1`	The adapter operation was executed successfully.
`-1`	An error occurred during connection establishment, disconnection, publishing or subscription.

Input

The adapter expects any input document for the operation `Publish`.

Output

The output is the same as the input.

Configuration file

To define broker, client, SSL/TLS configurations etc. a configuration file with a specified structure can be used, which is loaded at server startup. The configuration file `mqttconfig.xml` must be stored under the server path `X4DB/0`. If broker and client are defined in the configuration file, then only the values for `alias` and `cliendId` have to be specified in the adapter parameters.

The file contains the following information:

KeyStore	Path and password of the keystore that is used to establish a SSL connection to a broker. If this optional element is not defined, a trust manager is created that accepts all certificates.

When specifying the path, both absolute and relative paths to `X4DB/0` are allowed.

Example:

Relative path specification

```
<RootElement>
    <KeyStore password="p4ssw0rd">keystore.jks</KeyStore>
</RootElement>
```

Absolute path specification

```
<RootElement>
    <KeyStore password="p4ssw0rd">C:\Path\To\Keystore\keystore.jks</
KeyStore>
</RootElement>
```

Broker	MQTT server
	
	Relevant information is embedded in the attribute `alias`. This attribute must be unique. The broker contains the following elements:
	
	• `Server`: Broker URL; The protocols TSL and TCP are supported.
	  • *`ssl://host:port:`* The broker is connected via TSL protocol.
	  • Host is replaced by the host server and port by the port server.
	  • *`tcp://host:port`*: The broker is connected via TCP protocol (without SSL encryption).
	  Host is replaced by the host server and port by the port server.
	• `ClientId`: Unique client ID with the following attributes:
	  • `mode`: Type of the connection
	  • *`ALWAYS_ONLINE_ON_STARTUP`* (default): Adapter connects at server startup and holds connection all the time.
	  • *`ALWAYS_ONLINE_ON_FIRST_PUBLISH`*: The connection is established at the first request (Publish) and is held all the time.
	  • *`ONLINE_ON_DEMAND`*: The adapter connects if no connection exists and has an additional `timeout` attribute that is only valid in this mode. The attribute disconnects the connection after the specified time (in seconds) after the last message was sent.
	  If `mode` has an invalid value or is empty, the default value `ALWAYS_ONLINE_ON_STARTUP` is presumed.
	  • `timeout`: Timeout in seconds. This attribute is only valid for the mode `ONLINE_ON_DEMAND` and disconnects the connection after a specified time (in seconds) after the last message was sent.
	  If `timeout` is empty, the default value of 20 seconds is presumed.

Topic	Topic that can be subscribed to to receive the published message. It has the following attributes:
	• `filter`: Name of the topic filter
	• `qos`: Quality of service
	  • *`AT MOST ONCE`*: The message will be delivered only once or not at all. There will be no notification when the message has been delivered.
	  • *`AT LEAST ONCE`*: The message is always delivered at least once. If the sender does not receive a notification, the message with the DUP flag is delivered again until a notification is received.
	  • *`EXACTLY ONCE`*: The message must be stored locally at the sender and receiver until it has been processed.
	If the attribute `qos` is not defined, the default value is `AT_LEAST_ONCE`.

Sample configuration

```
<?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`: Allows to subscribe to a topic in an MQTT broker.

The MQTT protocol provides a simple method for sending messages via a publish/subscribe model. This makes it particularly suitable for messaging in the internet of things environment, such as low-power sensors or mobile devices like telephones, embedded computers or micro-controllers.

ⓘ The *MQTT Subscriber* is mostly used in combination with the *MQTT Publisher*.

Properties

`Operation`	Defines the operation executed by the adapter. **Possible values:** • `Subscribe`: Enables subscribing to a topic in an MQTT broker. • `Unsubscribe`: Enables unsubscribing to a topic in an MQTT broker.

Parameters

`Adapter`	Adapter main class (do not change!) **Possible values:** *de.softproject.integration.adapter.mqtt.MqttSubscriberAdapter*: Main class (default)
`Alias`	Broker alias **Possible values:** • Any string ⓘ The alias can be defined within the configuration file `mqttconfig.xml` as attribute of the element `Broker`.
`ClientId`	Identifier for the client that subscribes to the broker. **Possible values:** • If the Client ID is defined in the configuration file `mqttconfig.xml`: Any string • If the Client ID is not defined in the configuration file `mqttconfig.xml`: UUID ⓘ The Client ID can be defined within the configuration file `mqttconfig.xml` for each broker in the `ClientId` element.
`clientTimeout`	Timeout for ONLINE_ON_DEMAND connections (for the operation `Subscribe`). If `ConnectionMode` differs from `ONLINE_ON_DEMAND`, `ClientTimeout` is not taken into account. **Possible values:** Integer greater than `0`
`connectionMode`	Connection mode selected to create the Client. Depending on the selected value, the client connection is held or disconnected after a certain time (`ClientTimeout`). **Possible values:** • `ONLINE_ON_DEMAND`: The `ClientTimeout` parameter is required to configure the client. • `ALWAYS_ONLINE_ON_FIRST_PUBLISH`: The client stays connected as soon as an initial publishing happens. Therefore, the `ClientTimeout` parameter is not required. ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.

`stateful`	Defines if the state between client and broker is stateful. **Possible values:** `true` / `false` ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.
`keystorePath`	Path to the keystore. If no SSL/TLS configuration is required to connect to the broker, this parameter value may be empty. **Possible values:** File system path ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.
`keystorePassword`	Password for the keystore specified for `KeystorePath`. **Possible values**: Any string ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.
`Topic`	Name of the topic that the client subscribes to. **Possible values:** Any string ⓘ The topic can be defined within the configuration file `mqttconfig.xml` for each broker in the attribute `filter` of the element `Topic`.
`server`	Protocol, host and port to connect client and broker. **Possible values:** • `tcp://hostname:port`: If no SSL/TLS connection is used. • `ssl://hostname:port`: If a SSL/TLS connection is used. ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.

`InputFormat`	Input format of the adapter **Possible values:** • `BYTES` • `BASE64_STRING` • `BASE64_XML`
`loginUsername`	Username that is used to connect to a broker that requires credentials. **Possible values:** Username ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.
`loginPassword`	Password that is used to connect to a broker that requires credentials. **Possible values:** Password ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.
`QoS`	Quality of Service, for receiving messages when subscribing to a topic. **Possible values:** • `AT_MOST_ONCE` • `AT_LEAST_ONCE` (default) • `EXACTLY_ONCE` ⓘ This parameter is not required if `alias` and `clientId` are defined in the configuration file or the values for `alias` and `clientId` have already been used to establish a connection.
`process`	Path to the process that is executed each time a message of the subscribed topic is received. **Possible values:** Process path in the format *`x4db://User/process.wrf`*

Status values

`1`	The adapter operation was executed successfully.
`-1`	An error occurred during connection establishment, disconnection, publishing or subscription.

Input

The adapter expects no input document.

Output

The output is the same as the input.

Configuration file

To define broker, client, SSL/TLS configurations etc. a configuration file with a specified structure can be used, which is loaded at server startup. The configuration file `mqttconfig.xml` must be stored under the server path `X4DB/0`. If broker and client are defined in the configuration file, then only the values for `alias` and `cliendId` have to be specified in the adapter parameters.

The file contains the following information:

KeyStore	Path and password of the keystore that is used to establish a SSL connection to a broker. If this optional element is not defined, a trust manager is created that accepts all certificates.

When specifying the path, both absolute and relative paths to `X4DB/0` are allowed.

Example:

Relative path specification

```
<RootElement>
    <KeyStore password="p4ssw0rd">keystore.jks</KeyStore>
</RootElement>
```

Absolute path specification

```
<RootElement>
    <KeyStore password="p4ssw0rd">C:\Path\To\Keystore\keystore.jks</
KeyStore>
</RootElement>
```

Broker	MQTT server
	
	Relevant information is embedded in the attribute `alias`. This attribute must be unique. The broker contains the following elements:
	
	• `Server`: Broker URL; The protocols TSL and TCP are supported.
	  • *`ssl://host:port`* : The broker is connected via TSL protocol.
	  • ℹ Host is replaced by the host server and port by the port server.
	  • *`tcp://host:port`* : The broker is connected via TCP protocol (without SSL encryption).
	  ℹ Host is replaced by the host server and port by the port server.
	• `ClientId`: Unique client ID with the following attributes:
	  • `mode`: Type of the connection
	  • *`ALWAYS_ONLINE_ON_STARTUP`* (default): Adapter connects at server startup and holds connection all the time.
	  • *`ALWAYS_ONLINE_ON_FIRST_PUBLISH`* : The connection is established at the first request (Publish) and is held all the time.
	  • *`ONLINE_ON_DEMAND`* : The adapter connects if no connection exists and has an additional `timeout` attribute that is only valid in this mode. The attribute disconnects the connection after the specified time (in seconds) after the last message was sent.
	  ℹ If `mode` has an invalid value or is empty, the default value `ALWAYS_ONLINE_ON_STARTUP` is presumed.
	  • `timeout`: Timeout in seconds. This attribute is only valid for the mode `ONLINE_ON_DEMAND` and disconnects the connection after a specified time (in seconds) after the last message was sent.
	  ℹ If `timeout` is empty, the default value of 20 seconds is presumed.

Topic	Topic that can be subscribed to to receive the published message. It has the following attributes:
	
	• `filter`: Name of the topic filter
	• `qos`: Quality of service
	  • *`AT MOST ONCE`* : The message will be delivered only once or not at all. There will be no notification when the message has been delivered.
	  • *`AT LEAST ONCE`* : The message is always delivered at least once. If the sender does not receive a notification, the message with the DUP flag is delivered again until a notification is received.
	  • *`EXACTLY ONCE`* : The message must be stored locally at the sender and receiver until it has been processed.
	ℹ If the attribute `qos` is not defined, the default value is `AT_LEAST_ONCE`.
	For each topic a process is defined that is executed when a message is received. The element Process has the following attributes:
	• `url`: Process path in the format `x4db://User/process.wrf`

Sample configuration

```
<?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` : Establishes a connection to an OPC server and retrieves and sets values.

Properties

`Operation`	Defines the operation executed by the function adapter **Possible values:** *`execute`* : Execute the function adapter

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.adapter.opc.OpcAdapter`* : >Main class (Default)

`Host`	Host name or IP address of the OPC servers
`Domain`	Name of the OPC server's Windows domain
`User`	User name, to access the OPC server
`Password`	Password of the user
`ClsId`	Class ID of the OPC server software
`ProgID`	Windows program ID

Status values

`1`	The operation was executed without a technical error
`-1`	The operation was finished with a technical error The operation was possibly executed partially; More detailed information on the error cause can be found in the server log

Configuration

Within the folder `X4DB/0` a file `opc.xml` must be created, which has the following structure:

```
<opc_config>
   <Connection>
       <Host></Host>
       <Domain></Domain>
       <User></User>
       <Password></Password>
       <ClsId></ClsId>
   </Connection>
   <Group name=''>
       <Item name=''/>
       <Item name=''/>
   </Group>
</opc_config>
```

The following applies to each element `<Group>` and `<Item>`:

- It can be created any number of groups with any number of items
- A group is a collection of items assembled by the user that should be treated together
- The group name can be chosen individually
- The item name must match its name on the OPC server
- Read operations are always performed on the entire group
- An item can be in several groups at the same time
- When writing, it can only be written on the items that are in the same group

Input

The adapter expects a specific input XML structure. These input XML structures can be generated dynamically via XSLT mappings depending on the processed XML data.

```
<OPC>
   <read>
        <Group name=""/>
        <Group name=""/>
   </read>
   <write>
       <Group name="">
           <Item name="">...</Item>
           <Item name="">...</Item>
       </Group>
    </write>
    <copy>
       ...
   </copy>
</OPC>
```

Explanation

The following applies to each element `<read>`:

- For the element `read` any number of attributes can be defined, which are copied to the output
- A `read` element can contain any number of `Group`-Elemente enthalten

The following applies to each element `<write>`:

- For the element `write` any number of attributes can be defined, which are copied to the output
- A `write` element can contain any number of `Group` elements with any number of `Item` elements

The following applies to each element `<copy>`:

- For the element `copy` any number of attributes can be defined, which are copied to the output
- The `copy` element can contain any number of XML elements with any content; These elements are copied to the output

Output

ⓘ For all commands the respective attributes are copied to the container elements

Command: read

```
<OPC>
 <read>
 <Group name="">
 <Item name="">...</Item>
 <Item name="">...</Item>
 </Group>
 </read>
</OPC>
```

Command: write

The command will be executed, but no output will be generated:

```
<OPC>
 <write/>
</OPC>
```

Command: copy

```
<OPC>
 <copy> <!-- XML structure from the input document --> </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`: Connects to an OPC UA Server to read and write data from/to a machine in a Technical Process.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `GetServerConfiguration`: Calls the server configuration • `ListNodes`: Lists the available nodes • `ListNodesAttributtesId`: Lists the attributes for each node • `Read`: Reads out values and the status • `ReadHistory`: Reads out historical values and status • `Write`: Writes the attributes of a node

Parameter

`Adapter`	Adapter main class (do not change!) **Possible values:** *de.softproject.integration.adapter.opcua.OpcUaPublisherAdapter*: Main class (default)
`url`	Server URL **Possible values:** Any valid URL, e.g. *`opc.tcp://SP-WS-204.softproject.local:53530/OPCUA/SimulationServer`*
`user`	Username to connect to the server **Possible values:** Any string with the user name, e.g. *`myUser`*
`password`	Password to connect to the server **Possible values:** Any string with the password, e.g. *`myPassword`*
`securityMode`	Security Mode and Server Policies **Possible values:** • `NONE` • `BASIC256_SIGN` • `BASIC256_SIGN_ENCRYPT` • `BASIC128RSA15_SIGN` • `BASIC128RSA15_SIGN_ENCRYPT` • `BASIC256SHA256_SIGN` • `BASIC256SHA256_SIGN_ENCRYPT`
`autoReconnect`	Specifies whether the client should automatically reconnect to the server or not.
`nodeIndex`	Node index **Possible values:** Any integer with the node index, e.g. *`1234`*

`nodeIdentifier`	Node identifier **Possible values:** Any string with the node identifier, e.g. *myNode*
`nodeIdentifierType`	Type of the node identifier **Possible values:** • `String` • `Numeric`
`historyOption`	Options to retrieve the history of a specific node **Possible values:** • `AT_TIMES` • `AVERAGE` • `COUNT` • `DELTA` • `END` • `MAXIMUM` • `MAXIMUM_ACTUAL_TIME` • `MINIMUM` • `MINIMUM_ACTUAL_TIME` • `RANGE` • `RAW_DATA` • `START` • `WORST_QUALITY`
`from`	Start date with the format `yyyy-MM-dd hh:mm:ss` **Possible values:** Any date with the appropriate format, e.g. *2018-01-01 00:00:00*
`to`	End date with the format `yyyy-MM-dd hh:mm:ss` **Possible values:** Any date with the appropriate format, e.g. *2018-12-31 00:00:00*
`interval`	Interval value in seconds **Possible values:** Any whole time with the interval in seconds, e.g. *3000*
`arrayDelimiter`	Character to separate the values within an array **Possible values:** Any character (default `,`)

Status values

`1`	The adapter operation was executed successfully
`-1`	An error occurred while executing the adapter

Input

For the `Read` and `Write` operations, the adapter expects a specific XML document with the following elements:

- `nodeIndex` (Integer)
- `nodeIdentifier` (String)
- `nodeIdentifierType` (String , Numeric)
- `attributeId` (Integer)

- Use the `ListNodesAttributtesId` operation to call the node attribute ID.
- You specify arrays by enclosing the values in square brackets, for example [997, 998, 999].

Example input for the READ operation

```
<?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>
```

Example input for the WRITE operation without 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>
```

Example input for the WRITE operation with 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

The adapter outputs a specific XML document with the result:

Sample output for the READ operation

```
<?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>
```

Example output for the WRITE operation

```
<?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`: Establishes a connection to an OPC UA Server and subscribes or cancels data changes or events.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `Subscribe`: Subscribes to data changes or events according to the input XML document. • `UnSubscribe`: Cancels existing subscriptions for a specific client

Parameter

`Adapter`	Adapter main class (do not change!) **Possible values:** *de.softproject.integration.adapter.opcua.OpcUaSubscriberAdapter*: Main class (default)
`clientAlias`	Alias to identify the connection settings **Possible values:** Any string with the client alias
`process`	X4DB path to a Technical Process that is executed after each subscriber reception. **Possible values:** Any valid X4DB path, e. g. *`x4db://1/SampleProject/SampleProcess.wrf`*
`url`	Server URL **Possible values:** Any valid URL, e. g. *`opc.tcp://SP-WS-204.softproject.local:53530/OPCUA/SimulationServer`*
`user`	Username to connect to the server **Possible values:** Any string with the user name, e.g. *`myUser`*
`password`	Password to connect to the server **Possible values:** Any string with the password , e. g. *`myPassword`*
`security mode`	Security Mode and Server Policies **Possible values:** • `NONE` • `BASIC256_SIGN` • `BASIC256_SIGN_ENCRYPT` • `BASIC128RSA15_SIGN` • `BASIC128RSA15_SIGN_ENCRYPT` • `BASIC256SHA256_SIGN` • `BASIC256SHA256_SIGN_ENCRYPT`

`autoReconnect`	Specifies whether the client should automatically reconnect to the server or not.

Status values

`1`	The adapter operation was executed successfully
`-1`	An error occurred while executing the adapter

Input

Operation `Subscribe`:

- *Complete example with different subscription types:*

Input for the Subscribe operation

```
<?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>
```

- *Subscribe to data value change:*

  To subscribe to a data value change, a `Subscription` node must be created with the following mandatory attributes:
  - `type` to define the type of subscription
  - `nodeIndex`, `nodeIdentifier` to identify the OPC UA node from which the changes are subscribed to
  - `nodeIdentifierType` to specify the data type of the node identifier.

  The `Subscription` node can contain the following optional attributes:
  - `samplingInterval`: Value in milliseconds with which the OPC UA server samples the data source (PLC).
  - `publishingInterval` Value in milliseconds with which the OPC UA server sends the data to the client (e.g. X4).

```
<?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>
```

- *Subscribe to events without specifying fields:*

```
<?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
```

- *Subscribe to events and specify fields:*

```
<?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>
```

- *Subscribe to events and define filters for the specified fields:*

  It is possible to define conditions for the attributes that will be received on each event

reception. For this purpose, a corresponding number of `Field` nodes must be created. Each `Field` node contains the elements `Literal` and `Operand`.

Subscribe event with filters

```
<?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>
```

Note:

- The *name* attribute within `Field` is required.
- `Operand` can have the following values:
  - `LESS_THAN`
  - `LESS_THAN_OR_EQUALS`
  - `GREATER_THAN`
  - `GREATER_THAN_OR_EQUALS`
  - `EQUALS`
  - `NOT`
  - `LIKE`
- `Literal` contains the various literal values to be included within the expression.

Operation `Unsubscribe`

The `UnSubscribe` operation does not require any input, since a cancellation of the subscription is performed. Therefore, only the alias of the client must be specified in the `clientAlias` parameter.

Output

Operation `Subscribe`

The output structure depends on the type of receive message.

- When an event is received with fields (with or without conditions) configured in the subscription, the response structure resembles the following sample output:

Output for event responses

```
<?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>
```

Specifying conditions results in more or fewer events being received. However, once an event is received, the response contains the fields specified in the subscription.

- If an event is received without field specifications, the response structure resembles the following sample output:

Output for event responses

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Message type="EVENT"/>
```

- If a data value change is received, the response structure resembles the following sample output:

Output for data value changes

```
<?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`

The `UnSubscribe` operation has no output because the status `1` means that the operation was executed successfully.

Alive Listener Configuration

There is an Alive Listener that allows to configure actions for each live notification of a subscription.

Processes can be defined for the following situations:

- `After create`: The subscription has been (re)created on the server. This is the case when the subscription timed out due to a communication interruption and needs to be recreated after reconnection.
- `Alive`: The server confirmed that the connection was active, although there were no changes to send.
- `Life timeout`: The server has not confirmed that the connection is active and the life timeout has been exceeded.
- `Timeout`: The server has not confirmed that the connection is active and the maximum keepalive count has been exceeded.

The actions are defined as part of the input XML document as follows:

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`: Allows to connect to *Siemens SIMATIC S7* systems as PLCs from within Technical Processes. Data can be read from the controller as well as transmitted to it. An XML input structure allows to determine which data should be read or written. *SIMATIC* systems of the series S7-200, 300, 400 can be connected.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • `Read`: Read data from PLC • `Write`: Write data to PLC

Parameters

Adapter	Adapter main class (do not change!) **Possible values:** *`de.softproject.integration.adapter.s7.S7 Connector`:* Main class (default)
host	Address to reach the PLC in the network **Possible values:** IP address or Host name
rack	Rack to be addressed on the PLC **Possible values:** Rack number (ask the PLC developer)

slot	Slot to be addressed on the PLC **Possible values:** Slot number (ask the PLC developer)

Status values

1 (ok)	Successful communication (reading or writing) with the PLC
-1 (error)	Error during communication (reading or writing) with the PLC; An error message is returned

Input

This adapter expects an XML document with the following input structure:

Input: Operation Read

```
<?xml version="1.0" encoding="UTF-8"?>
<Read dbnum="[DBNUM]" bytes="[LENGTH]" offset="[OFFSET]">
    <FELDNAME s7type="[TYPE]" s7size="[LENGTH]" />
    ...
</Read>
```

Parameters:

- `dbnum`: DB number to be read from (ask the PLC developer)
- `bytes`: Number of the bytes to be read
- `offset`: Offset in bytes from which is to be read
- The elements `FELDNAME` are freely selectable
- `s7type`: Type of the fielt to be read;
  Possible values:
  - `Array`
  - `Bit` (**Note:** only reads 1 bit, therefore 8 fields must be always specified!)
  - `Byte`
  - `DateAndTime`
  - `Date` (in format `yyyy-MM-dd'T'HH:mm:ss`)
  - `Integer`
  - `Long`
  - `Real`
  - `String`
  - `Time` (as Long)
- `s7size`: Length in bytes to be read for the field. Only necessary for the types array and string

Input: Operation Write

```
<?xml version="1.0" encoding="UTF-8"?>
<Write dbnum="DBNUM" bytes="LENGTH" offset="OFFSET">
    <FELDNAME s7type="String" s7size="12">[VALUE]</AUFNR>
    ...
</Write>
```

Parameters:

- see operation `Read`
- The element's `VALUE` must syntactically match the type

Output

This connector outputs an XML document with the following structure:

- Operation `Read`: The output corresponds to the input; If successful, the elements are filled with the appropriate values
- Operation `Write`: The output corresponds to the input

10.14 Message Oriented Middleware

- Apache Kafka
- AMQP Publisher/Subscriber
- JMS
- Websphere MQ

10.14.1 Apache Kafka

Apache Kafka enables data streams to be stored and processed over a distributed streaming platform. Various interfaces allow data to be written to Kafka clusters, data to be read or imported or exported from third-party systems. Due to its high fault tolerance and scalability, Apache Kafka is suitable for large amounts of data and applications, e.g. in the big data environment.

> ⓘ More information about using Apache Kafka in a local environment can be found here.
> More information about configuring and integrating TLS certificates in Apache Kafka Cluster can be found here.

A configuration file is required in the X4 server installation to use the two adapters Apache Kafka Publisher or Apache Kafka Subscriber. This configuration file must be called `apachekafka-config.xml` and located in the `X4DB/0` folder. When the X4 server is started, all information from this file is loaded.

The following XML file shows all required elements and attributes in the correct structure:

Example file 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>
```

The most important elements and attributes are explained in more detail below:

Element / Attribute	Description
`Cluster`	*Must be unique*. Defines the parameters required to configure the connection to a particular Kafka cluster. The `id` attribute contains the information required for the adapter parameter `clusterId`.
`server`	URL of the cluster. Supports the TCP protocol. **Possible values:** `hostname:port`
`SslConfiguration`	Defines the SSL/TLS configuration parameters.
`privateKeyPassword`	Password of the private key in the keystore file.
`KeyStore`	Defines where the keystore to be used is located and which password is stored.
`TrustStore`	Defines where the trust store to be used is located and which password is stored.
`ClientId`	Information about the client that is used to publish messages on a specific topic.
`groupId`	Defines the group ID used to group the connections.
`autoCommitIntervalMS`	Defines the frequency of commits in milliseconds. ⓘ This attribute is required for the adapter Apache Kafka Subscriber.
`Topic`	Topic in which the messages are published.
`filter`	Name of the topic

ⓘ The values for `KeyStore`, `TrustStore` and `privateKeyPassword` can also be set via adapter parameters. Values set via adapter parameters override the values from `apachekafka-config.xml`.

10.14.1.1 Apache Kafka Publisher

`Apache Kafka Publisher`: Enables to publish messages in a specific Apache Kafka Topic.
Properties

`Operation`	Defines the operation executed by the adapter. **Possible values:** • `Publish`: Publishes a message in a specific topic that is specified in an XML input.

Parameter

`Adapter`	Adapter main class (do not change!) **Possible values:** *de.softproject.integration.adapter.apachekafka.ApacheKafkaPublisherAdapter*: Hauptklasse (default)
`trustStorePath`	Defines where the trust store to be used is located.
`trustStorePassword`	Password for the trust store
`keyStorePath`	Defines where the key store to be used is located.
`keystorePassword`	Password for the key store
`privateKeyPassword`	Password of the private key in the keystore file
`username`	User name for SASL/PLAIN authentication at the Kafka server ⚠ If you use this parameter, the configuration properties `sasl.mechanism = PLAIN` and `security.protocol = SASL_SSL` are set.
`password`	Password for SASL/PLAIN authentication at the Kafka server ⚠ If you use this parameter, the configuration properties `sasl.mechanism = PLAIN` and `security.protocol = SASL_SSL` are set.
`clusterId`	ID of the cluster in which the message is published **Possible values:** Cluster ID ⓘ The ID of the cluster can be found in the file `apachekafka-config.xml`. It is located in the `id` attribute of the `Cluster` element.

`ClientId`	ID of the client with which the message is published **Possible values:** Client ID ⓘ The ID of the cliente can be found in the file `apachekafka-config.xml`. It is located in the `id` attribute of the `Client` element.
`topic`	Name of the topic in which the message is published **Possible values:** String consisting of alphanumeric ASCII characters, `.` (dot), _ (underscore) and – (hyphen)
`messageDataType`	Data type of the message(s) to be published. **Possible values:** • `BYTES`: Base64 encoded value • `BYTEARRAY`: Base64 encoded value • `DOUBLE`: Floating-point number • `FLOAT`: Floating-point number • `INTEGER`: Integer • `LONG`: Integer • `SHORT`: Integer • `STRING` (default): String • `UUID`: Java UUID ⚠ Messages with different data types can be published in a topic. To read the messages with the Apache Kafka Subscriber Adapter, the data type must be known and set correctly. For example, a message of the data type `STRING` cannot be read with the data type setting `LONG`. Therefore, all messages in a topic should be of the same data type.

keyDataType	Data type of the key
	
	Possible values:
	
	• NONE (default): This value is used if no key is defined.
	• BYTES: Base64 encoded value
	• BYTEARRAY: Base64 encoded value
	• DOUBLE: Floating-point number
	• FLOAT: Floating-point number
	• INTEGER: Integer
	• LONG: Integer
	• SHORT: Integer
	• STRING (default): String
	• UUID: Java UUID
	
	Sample key definition
	
	`<Messages>`
	`<Message key="055b93a8-4eda-41d8-8b73-b852a1d4da1e">055b93a8-4eda-41d8-8b73-b852a1d4da1e</Message>`
	`</Messages>`
	
	ⓘ The following four cases can be distinguished:
	
	• A key with a data type other than NONE has been defined: The message is sent with a key in the specified data type.
	• A key with the NONE data type has been defined: The message is sent without a key and a warning is issued in the server log.
	• No key has been defined and the data type of the key has a value other than NONE: The message will not be sent and an error message will be issued.
	• No key has been defined and the data type of the key is NONE: The message will be sent without a key.

ⓘ The values of the `trustStorePath`, `trustStorePassword`, `keyStorePath`, `keyStorePassword` and `privateKeyPassword` parameters can also be defined in the `apachekafka-config.xml` file (see Apache Kafka).

⚠ The information about the adapter configuration overwrites the information in the `apachekafka-config.xml`. However, **all** parameters must be set for this. If not all parameters are set via the adapter configuration, the files from the `apachekafka-config.xml` are used.

⚠ If adapter parameters are used to override the SSL configuration of a particular client, then these values are retained until they are overridden again or the server is restarted.

Status values

`1`	The adapter operation was executed successfully.
`-1`	An error occurred during the execution of the adapter.

Input

The adapter expects a predefined XML structure as input. The structure depends on the data type of the message.

- Data type `STRING`:

```
<Messages>
    <Message>
        <!-- This is a message. -->
    </Message>
    <Message>
        <!-- This is another message. -->
    </Message>
</Messages>
```

- Data type `BYTES`:

```
<Messages>
    <Message>ZXN0byBlcyB1biBmaWNoZXJvIGRlIHRleHRv</Message>
</Messages>
```

- Data type `NUMERIC`:

```
<Messages>
    <Message>5428</Message>
</Messages>
```

Messages with and without headers can be mixed:

```
<?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

If the execution is successful, the input is output.

10.14.1.2 Apache Kafka Subscriber

`Apache Kafka Subscriber`: Enables to subscribe messages in a specific topic, unsubscribe or read the last message.
Properties

`Operation`	Defines the operation executed by the adapter. **Possible values:** • `Subscribe`: Subscribes to a topic to receive messages. An existing client from the file `apachekafka-config.xml` is used. ⚠ Only a client that is not yet used for another topic can be used for a subscription. A client cannot subscribe to multiple topics. **Required parameters:** `clusterId`, `clientId`, `topic`, `process` • `Unsubscribe`: Disconnects client and topic and unsubscribes from a specific topic. **Required parameters:** `clusterId`, `clientId` • `Last offset`: Returns the offset of a specific client for its subscribed topic. With the offset past messages can be received again. ⚠ If this operation is used for a client that has not subscribed to a topic, an empty result is returned. **Required parameters:** `clusterId`, `clientId`, `topic`

Parameter

`Adapter`	Adapter main class (do not change!) **Possible values:** *de.softproject.integration.adapter.apachekafka.ApacheKafkaSubscriberAdapter*: Main class (default)
`trustStorePath`	Defines where the trust store to be used is located.
`trustStorePassword`	Password for the trust store
`keyStorePath`	Defines where the key store to be used is located.
`keystorePassword`	Password for the key store
`privateKeyPassword`	Password of the private key in the keystore file
`username`	User name for SASL/PLAIN authentication at the Kafka server ⚠ If you use this parameter, the configuration properties `sasl.mechanism = PLAIN` and `security.protocol = SASL_SSL` are set.

`password`	Password for SASL/PLAIN authentication at the Kafka server ⚠ If you use this parameter, the configuration properties `sasl.mechanism = PLAIN` and `security.protocol = SASL_SSL` are set.
`clusterId`	ID of the cluster with which the topic is subscribed **Possible values:** Cluster ID ⓘ The ID of the cluster can be found in the file `apachekafka-config.xml`. It is located in the `id` attribute of the `Cluster` element.
`ClientId`	ID of the client with which the topic is subscribed **Possible values:** Client ID ⓘ The ID of the cliente can be found in the file `apachekafka-config.xml`. It is located in the `id` attribute of the `Client` element.
`topic`	Name of the topic to be subscribed to **Possible values:** String consisting of alphanumeric ASCII characters, `.` (dot), _ (underscore) and – (hyphen) or comma-separated list of multiple strings to subscribe to multiple topics
`process`	Process that is executed after a message is received. **Possible values:** String consisting of alphanumeric ASCII characters, `.` (dot), _ (underscore) and – (hyphen)
`consumerPosition`	Offset with which messages are received when a topic is subscribed to. **Possible values:** Integer greater than `0` ⚠ This parameter only has an effect if a single topic is subscribed to. If several topics are subscribed to, this parameter has no effect.

`messageDataType`	Data type of the message(s) to be published. **Possible values:** • `BYTES`: Base64 encoded value • `BYTEARRAY`: Base64 encoded value • `DOUBLE`: Floating-point number • `FLOAT`: Floating-point number • `INTEGER`: Integer • `LONG`: Integer • `SHORT`: Integer • `STRING` (default): String • `UUID`: Java UUID ⚠ Messages with different data types can be published in a topic. To read the messages with the Apache Kafka Subscriber Adapter, the data type must be known and set correctly. For example, a message of the data type `STRING` cannot be read with the data type setting `LONG`. Therefore, all messages in a topic should be of the same data type.
`keyDataType`	Data type of the key **Possible values:** • `NONE` (default): No data type ⓘ If the subscriber has subscribed to a topic and the `keyDataType` has the value `NONE`, only messages without a key will be received. If the value is other than `NONE`, messages with all other data types are received. If the message has a key, the key is output as an attribute in the `Content` element. • `BYTES`: Base64 encoded value • `BYTEARRAY`: Base64 encoded value • `DOUBLE`: Floating-point number • `FLOAT`: Floating-point number • `INTEGER`: Integer • `LONG`: Integer • `SHORT`: Integer • `STRING` (default): String • `UUID`: Java UUID

ⓘ The values of the `trustStorePath`, `trustStorePassword`, `keyStorePath`, `keyStorePassword` and `privateKeyPassword` parameters can also be defined in the `apachekafka-config.xml` file (see Apache Kafka).

⚠ The information about the adapter configuration overwrites the information in the `apachekafka-config.xml`. However, **all** parameters must be set for this. If not all parameters are set via the adapter configuration, the files from the `apachekafka-config.xml` are used.

⚠ If adapter parameters are used to override the SSL configuration of a particular client, then these values are retained until they are overridden again or the server is restarted.

Status values

`1`	The adapter operation was executed successfully.
`-1`	An error occurred during the execution of the adapter.

Input

This adapter expects no input.

Output

The adapter outputs the following output depending on the operation executed:

- Operation `Subscribe`: No direct output is output for this operation. If a new message is received in a subscribed topic, a process is started with the received message as input. This message is provided in the following structure:

  Sample 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`: No output
- Operation `Last offset`:

Sample output operation Last offset

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Offsets>
    <Offset partitionId="0" value="17"/>
</Offsets>
```

In the output of this adapter operation, the value of `partitionId` shows the partition where the topic is located, and the value of `value` shows the offset to the last message read by the specified client.

10.14.2 AMQP Publisher/Subscriber

10.14.2.1 AMQP Publisher

`AMQP Publisher`: Publishes messages to an AMQP topic.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Publish`: Publish messages provided as input into an AMQP topic

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.amqp.AmqpPublisherAdapter*: Main class (default)
`storedConfig`	Identifier of the configuration provided in `amqpconfig.xml` **Possible values:** Any string containing the configuration identifier, e.g. *`queueEmployeesConfig`*
`host`	AMQP server host **Possible values:** Server host, e.g. *`localhost`*

`port`	Dedicated AMQP port in the server **Possible values:** AMQP port number, e.g. *`5672`*
`username`	Name of the user publishing the messages **Possible values:** Any string containing the user name, e.g. *`user`*
`password`	Password associated to the user **Possible values:** Any string containing the corresponding password, e.g. *`p4ssw0rd`*
`queue`	Queue where the message will be published **Possible values:** Any string containing the queue name, e.g. *`employees`*
`container`	Container where the queue is placed **Possible values:** Any string containing the container name, e.g. *`employeesContainer`*
`qos`	Quality of Service of the message **Possible values:** • `AT_MOST_ONCE`: The message is delivered at most once, or it is not delivered at all. • `AT_LEAST_ONCE`: The message is delivered at least once. • `EXACTLY_ONCE`: The message is delivered only once. The default value is `AT_MOST_ONCE`, if the parameter `qos` is not specified.
`useSsl`	Enable SSL encryption **Possible values:** `true` / `false` (default)
`useSasl`	Activate SASL Authentication **Possible values:** `true` / `false` (default)

`mechanism`	SASL mechanism of the remote endpoint. **Mögliche Werte:** • `PLAIN`: All data is exchanged in cleartext (Default). • `ANONYMOUS`: The service can be used for unauthenticated guest access. • `CRAM-MDS`: Prevents sending the password in cleartext. • `Digest-MDS`: Additional parameters can be used for integrity checks.
`durable`	Set the message as durable **Possible values:** `true` (default) / `false`
`priority`	Priority level of the message **Possible values:** Any number between `0` and `255`, e.g. *3*
`ttl`	Period of time in milliseconds when the message is considered alive **Possible values:** Any integer indicating the period of time in milliseconds, e.g. *4500*

Status values

`0`	The operation was executed successfully (no output is generated)
`-1`	An error occurred during the operation's execution (no output is generated)

Configuration

To provide the needed information a configuration file with a specified structure can be used, which is loaded at server startup. The configuration file `amqpconfig.xml` must be stored under the server path `X4DB/0`.

The file has the following structure:

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

The AMQP Publisher expects the following input structure, whereby only one element for each message is allowed:

Simple types:

Sample input: simple type

```
<Messages>
    <Message>
        <Element type="String" value="Good morning!!"/>
    </Message>
    <Message>
        <Element type="Int" value="18"/>
    </Message>
</Messages>
```

The following simple types are allowed: `Binary`, `Boolean`, `Byte`, `Char`, `Decimal32`, `Decimal64`, `Decimal128`, `Double`, `Float`, `Long`, `Short`, `String`, `Symbol`, `Timestamp` (following the pattern *`yyyy-[m]m-[d]d hh:mm:ss`*), `UByte`, `UInt`, `ULong`, `UShort`, `Uuid`.

Complex types:

- List:

  Sample input: List

  ```
  <Messages>
      <Message>
          <Element type="List">
              <Element type="Int" value="1"/>
              <Element type="Short" value="3"/>
              <Element type="Boolean" value="false"/>
          </Element>
      </Message>
  </Messages>
  ```

- Map:

Sample 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`: Manages subscriptions to an AMQP queue and defines the process to be executed.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Subscribe`: Subscribe to an AMQP queue and define the process to be executed ℹ Required parameters are `host`, `port`, `username`, `password`, `queue` and `process`. The parameters can also be provided via a configuration file, which is specified within the parameter `storedConfig`. • `Unsubscribe`: Unsubscribe from an AMQP queue ℹ Required parameters are `username` and `queue`. The parameters can also be provided via a configuration file, which is specified within the parameter `storedConfig`.

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.amqp.AmqpSubscriberAdapter*: Main class (default)

`storedConfig`	Identifier of the configuration provided in `amqpconfig.xml` **Possible values:** Any string containing the configuration identifier, e.g. *`queueEmployeesConfig`*
`host`	AMQP server host **Possible values:** Server host, e.g. *`localhost`*
`port`	Dedicated AMQP port in the server **Possible values:** AMQP port number, e.g. *`5672`*
`username`	Name of the user subscribing or unsubscribing to a topic **Possible values:** Any string containing the user name, e.g. *`user`*
`password`	Password associated to the user **Possible values:** Any string containing the corresponding password, e.g. *`p4ssw0rd`*
`queue`	Queue where the message is subscribed or unsubscribed **Possible values:** Any string containing the queue name, e.g. *`employees`*
`container`	Container where the queue is placed **Possible values:** Any string containing the container name, e.g. *`employeesContainer`*
`qos`	Quality of Service of the message **Possible values:** • `AT_MOST_ONCE`: The message is delivered at most once, or it is not delivered at all. • `AT_LEAST_ONCE`: The message is delivered at least once. • `EXACTLY_ONCE`: The message is delivered only once. The default value is `AT_MOST_ONCE`, if the attribute `qos` is not specified.

`useSsl`	Enable SSL encryption **Possible values:** `true` / `false` (default)
`useSasl`	Activate SASL Authentication **Possible values:** `true` / `false` (default)
`mechanism`	SASL mechanism of the remote endpoint. **Mögliche Werte:** • `PLAIN`: All data is exchanged in cleartext (Default). • `ANONYMOUS`: The service can be used for unauthenticated guest access. • `CRAM-MDS`: Prevents sending the password in cleartext. • `Digest-MDS`: Additional parameters can be used for integrity checks.
`process`	Path of the Technical Process executed as soon as a message is received **Possible values:** Process path, e.g. *`x4db://1/Project/Processes/process.wrf`*

Status values

`0`	The operation was executed successfully (no output is generated)
`-1`	An error occurred during the operation's execution (no output is generated)

Configuration

To provide the needed information a configuration file with a specified structure can be used, which is loaded at server startup. The configuration file `amqpconfig.xml` must be stored under the server path `X4DB/0`.

The file has the following structure:

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`: Exchanges messages via the *Java Message Service* queue of a message-oriented middleware product that serves as a JMS provider. The adapter supports sending and receiving messages to and from the JMS queue and can publish messages to JMS topics and subscribe to JMS topics.

Former designation up to version 4.5.1: *JMS Transfer AdapterFormer*
designation up to version 5.5.4: *JMS Connector*

Properties

Delete	(no function)
File name	(no function)
I/O Mode	(no function)
Operation	Determines which operation is performed on the messages in the JMS queue. **Possible values:** • `Send`: Sends data from the previous process step to the JMS queue. • `Receive`: Reads JMS message into the process • `List`: Outputs the number of messages in the queue as an XML document (if parameter `toXML` is set, the header information of the messages is output instead).

Parameter

object_name	Name of the JMS queue or JMS topic under which it is registered in the JNDI provider (Java Naming and Directory Interface). **Possible values:** Any string (e.g. *queue/A*)
connection_factory	Name of the *JMS Connection Factory* under which the message queue is registered in the JNDI (Java Naming and Directory Interface). **Possible values:** Any valid string (e.g. *java:/JmsXA*)

JNDIProperties	Additional properties of the JNDI provider **Possible values:** Any string with properties, separated by semicolons if there are several properties (e.g. *java.naming.factory.initial=org.jnp.inter faces.Naming ContextFactory;java.naming.provi der.url=localhost:1099*)
approach	Used messaging channel type **Possible values:** • `topic`: Point-to-point connection between a sender and a receiver, who normally removes the message from the message channel after receiving it. • `queue`: Topic-oriented publish/subscribe connection, where messages remain in the message channel to be retrieved by other recipients until the messages are replaced or expire (default).
timeout	Time limit for establishing connection in milliseconds **Possible values:** • Any integer • `0`: no restriction; Caution: If the queue is empty, then the process "stands" with this setting! • `10000`: Wait maximum 10 seconds (default)
filter	Filters when receiving messages via JMS message selectors **Possible values:** Any string with properties according to the pattern `<Property1>=<Value1>;<Property2>=<Value2>` (e.g. *`JMSPriority=5`*)
msg_property	Header attributes of the JMS message when sending messages **Possible values:** Any string with properties according to the pattern `<property1>=<value1>;<property2>=<value2>`.
headers_from_XML	Use header information of the input document **Possible values:** • `yes`: use header information • `no`: Do not use header information (default)

transacted	Execute this process step in a local transaction **Possible values:** • `yes`: Execute as local transaction • `no`: Do not execute a local transaction (default)
ack_mode	Acknowledgement mode (this regulates the handling of acknowledgements of receipt) **Possible values:** • `AUTO_ACKNOWLEDGE|Auto acknowledge`: Automatically acknowledge receipt of messages • `CLIENT_ACKNOWLEDGE|Client acknowledge`: Acknowledge receipt by a client. • `DUPS_OK_ACKNOWLEDGE|Dups ok acknowledge`: Allow multiple reception of a message
msg_priority	Message priority (*9* has the highest priority) **Possible values:** • Any integers from *0–9* • `4`: Priority 4 (standard)
msg_expiration_time	Validity period of messages (lifetime) in milliseconds; after the validity has expired, the message is obsolete and is automatically removed from the queue **Possible values:** • Any positive integer • `0`: no validity restriction (default)
msg_type	JMS message type **Possible values:** • `text`: Exchange data as character strings (e.g. as XML documents) (standard) • `byte`: byte field group for data exchange in native/binary formats • `map`: Send JMS MapMessages (Key-Value-Pairs)

subscription_name	Name of the subscription for permanent message subscriptions (if parameter approach has been set to *topic*). **Possible values:** Any string
durable_subscription	Dauer des JMS-Subscriptions **Possible values:** • `yes`: obtain permanently • `no`: obtain temporarily
ClientID	Name of the JMS-Clients **Possible values:** Any string
toXML	Output the JMS message including header information as an XML document (if the Operation property is set to *Receive*, the character set is adopted unchanged; *Text/XML* is then used as the MIME type). If you do not set the parameter toXML, then byte, stream or object messages are declared as MIME type *Application/Octet-stream*; text messages, on the other hand, are UTF-8 encoded and declared with MIME type *Text/Plain*. **Possible values:** • `yes`: Output XML document • `no`: Do not output an XML document (default)
base64	Output JMS message Base64-encoded (if parameter toXML is set) **Possible values:** • `yes`: Output message Base64-encoded • `no`: Do not output a Base64-encoded message (default)
JMSReplyTo	Response message destination queue
JMSCorrelationID	ID of the message to which the response refers (different parts of a message have the same JMSCorrelationID).
login	Username for the JMS
password	Password for the JMS

The JMS queues available by default in the application server are not suitable for productive use because of the database system used (HSQLDB, Derby). Despite transactional processing, data can be lost with these. Only use database systems that can be used productively (Oracle, MSSQL etc.) for message persistence of JMS queues; see http://docs.jboss.org/jbossas/docs/Server_Configuration_Guide/4/html/apbs04.html.

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

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`: Monitors an *IBM Websphere MQ Server* via commands which are handed over.

> **Dependencies**
> The adapter requires the chargeable library `com.ibm.mq.allclient.jar` within the `Classpath`.

Properties

`Operation`	Defines the operation executed by the function adapter **Possible values:** • *`Query`*: Executes the commands handed over within an input XML document and outputs them in an XML document

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.ibmmq.MQChannelChecker*: Main class (Default)
MonitorIp	IP address of the MQ server to be monitored **Possible values:** any valid string (e. g. *`127.0.0.1`*)
MonitorPort	Port of the MQ server to be monitored **Possible values:** any positive number (e. g. *`4711`*)
MonitorChannel	Name of the Channel to be monitored on the MQ server **Possible values:** any valid string (e. g. *`Channel`*)

Status values

`1 (`*`successful`*`)`	All commands have been executed successfully
`-1 (`*`failed`*`)`	The request failed due to a technical error

Input

This adapter expects an XML document as input, which has the following structure:

```
<Commands>
    <{Command} />
</Commands>
```

The following commands are supported:

- *`InquireQueue`*
- *`InquireChannelNames`*
- *`InquireChannelStatus`*

Example for the three commands:

```
<Commands>
    <InquireQueue queue="{value}" />
    <InquireChannelNames channel="{value}" />
    <InquireChannelStatus channel="{value}" />
</Commands>
```

Output

The adapter expands within the output XML document the command element by the corresponding result.

Example

```
<Commands>
    <InquireQueue queue="{value}">
        <Message>
            <Value name="{value}">{value}</Value>
        </Message>
    </InquireQueue>
    <InquireChannelNames channel="{value}">
        <NoResult />
    </InquireChannelNames>
    <InquireChannelStatus channel="{value}">
        <Message>
            <Value name="{value}">{value}</Value>
            <Value name="{value}">{value}</Value>
        </Message>
    </InquireChannelStatus>
</Commands>
```

10.14.4.2 MQ Series Transfer

`MQ Series Transfer`: Sends and receives messages via an I*BM Websphere MQ* (*MQ Series*) message queue and supports MQ-specific features such as *Dynamic Queues*.

Former name until version 4.5.1: *MQ Series Transfer Adapter*

Properties

Delete	(no function)
File name	(no function)
I/O Mode	(no function)
Operation	Determines which operation is executed with the message queue. **Possible values:** • *`Send`*: Send data from the previous process step to the *MQ Series* queue. • *`Receive`*: Read message into the process • *`Commit`*: Complete transaction • *`Rollback`*: Cancel transaction and undo all changes • *`Create Queue`*: Create queue • *`Destroy Queue`*: Delete queue • `Clear`: Reset connection to MQ Series

Parameter

alias	Name of the parameter group for external configuration of the adapter via the `QueueConfig.xml` file (in `x4.ear` in the `conf` subfolder). Values specified there have a higher priority than the parameters of the Transfer Adapter process module.
toXML	If property Operation was set to *`Receive`*: Output the result of the process step as an XML document; if the `MQMessage.format` parameter is also set to *`NONE`*, the message is Base64-encoded. **Possible values:** • *`true`*: Generate XML document containing header and body of the message with all properties (see example) • *`false`*: Generate an MQ message as a result of the process step (default).

ccdtUrl	URL to a *Client Channel Definition Table (CCDT)* ? Unknown Attachment There are four different ways to define the `ccdtUrl`: 1. Via the Java arguments `-DMQCHLLIB` and `-DMQCHLTAB` when starting the *X4 Server* **Possible values:** • `-DMQCHLLIB:` Path to the table file • `-DMQCHLTAB:` *`AMQCLCHL.TAB`* (IBM default name) 2. By the Java argument `-Dx4.mqseries.ccdtUrl` Example: `-Dx4.mqseries.ccdtUrl=file:///`*`Path_to_Table`*`/AMQCLCHL.TAB` 3. Via the `ccdtUrl` property in *`QueueConfig.xml`* 4. Via the `ccdtUrl` parameter in the adapter itself The above priority is applied.
RemoteQueueManager.name	Name of the queue manager **Possible values:** • Any valid string • *{EMPTY}*: Explicitly no value (empty string)
MQEnviroment.hostname	Hostname of the addressed MQ server; applies system-wide within the entire Java VM for all following operations (the parameter is only needed for local tests on development systems, otherwise leave it empty!) **Possible values:** • Any string (e.g. *MQSR2402*) • *{EMPTY}*: Explicitly no value (empty string)
MQEnviroment.channel	Name of the channel;applies system-wide within the entire Java VM for all following operations (only needed for local tests on development systems, otherwise leave empty!) **Possible values:** • Any string (e.g. *MQSR2402.CLS25740.DT*) • *{EMPTY}*: Explicitly no value (empty string)

MQEnviroment.CCSID	CCSID (Charset) of all messages for the entire Java VM; applies system-wide within the entire Java VM for all subsequent operations, therefore use with care (should not be used!) **Possible values:** • Any CCSID (e.g. *1208*) • *{EMPTY}*: Explicitly no value (empty string)
MQEnviroment.userID	User ID for authentication of the *WebSphere MQ client* ; valid system-wide within the whole Java VM for all following operations (only needed for local tests on development systems, otherwise leave empty!)
MQEnviroment. localAddressSetting	MQ-specific parameter; applies system-wide within the entire Java VM for all following operations (only needed for local tests on development systems, otherwise leave empty!) **Possible values:** • Any valid string • *{EMPTY}*: Explicitly no value (empty string)
MQEnviroment.password	Password for authentication of the *WebSphere MQ client* ; valid system-wide within the whole Java VM for all following operations (only needed for local tests on development systems, otherwise leave empty!) **Possible values:** • Any valid string • *{EMPTY}*: Explicitly no value (empty string)
MQEnviroment.port	Port of the MQServer; applies system-wide within the entire Java VM for all following operations (only needed for local tests on development systems, otherwise empty). **Possible values:** • Any integer (e.g. *1414*) • *{EMPTY}*: Explicitly no value (empty string)

MQEnviroment.properties.MQC.SSL_CIPHER_SUITE_PROPERTY	`CipherSuite` configuration for the current connection. The `CipherSuite` entered must correspond to the CipherSpec used: **`CipherSpec` and the corresponding `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*: *SSL_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* ⓘ Additional information on `CipherSuite` configuration can also be found here (Chapter 4) **Possible values:** `CipherSuite` as valid string, see above.
MQEnviroment.properties.MQC.TRANSPORT_PROPERTY	Setting for the current connection **Possible values:** *MQC.TRANSPORT_MQSERIES*: Only allowed value
MQEnviroment.properties.MQC.CCSID_PROPERTY	CCSID (Charset) for the current connection **Possible values:** Any integer (e.g. *1208*)

MQEnviroment.properties.MQC.USER_ID_PROPERTY	User ID for the current connection **Possible values:** Any string
MQEnviroment.properties.MQC.CHANNEL_PROPERTY	Name of the channel for the current connection **Possible values:** Any string (e.g. *MQSR2402.CLS25740.DT*)
MQEnviroment.properties.MQC.CONNECT_ OPTIONS_PROPERTY	Connection options for the current connection **Possible values:** Any string
MQEnviroment.properties.MQC.HOST_NAME_PROPERTY	Host name of the addressed MQ server for the current connection. **Possible values:** Any string (e.g. *MQSR2402*)
MQEnviroment.properties.MQC.LOCAL_ADDRESS_PROPERTY	Local address for the current connection **Possible values:** Any string
MQEnviroment.properties.MQC.ORB_PROPERTY	ORB property for the current connection **Possible values:** Any string
MQEnviroment.properties.MQC.PASSWORD_PROPERTY	Password property for the current connection setup to authenticate the *WebSphere MQ client* (corresponds to parameter MQEnviroment.password). **Possible values:** Any string
MQEnviroment.properties.MQC.PORT_PROPERTY	Port for the current connection **Possible values:** Any string
MQEnviroment.properties.MQC.RECEIVE_EXIT_PROPERTY	*Receive Exit* property for the current connection **Possible values:** Any string
MQEnviroment.properties.MQC.SECURITY_EXIT_PROPERTY	*Security Exit* property for the current connection **Possible values:** Any string
MQEnviroment.properties.MQC.SEND_EXIT_PROPERTY	*Send Exit* property for the current connection **Possible values:** Any string

MQQueueManager.name	Name of the QueueManager. Only necessary for local tests on development systems, otherwise empty.
MQQueue.name	Queue name
MQQueue.dynamic QueueName	Name of the session queue for this message (Should not be used in the future)
MQQueue.alternate UserId	Alternative user ID to check access to the queue when it is open; this parameter cannot be set while the object is open
MQQueue.openOptions. MQC.MQOO_OUTPUT	(Must be set for write access)
MQQueue.openOptions. MQC.MQOO_FAIL_ IF_QUIESCING	Allows the MQ Manager to disconnect the listener (e.g. at shutdown).
MQQueue.openOptions. MQC.MQDL_SUPPORTED	MQ specific parameter
MQQueue.openOptions. MQC.MQOO_ALTERNATE_ USER_AUTHORITY	MQ specific parameter
MQQueue.openOptions. MQC.MQOO_ BIND_AS_ QDEF	Apply queue settings for persistence etc. (should always be set) **Possible values:** • *yes*: Apply queue settings • *no*: Do not apply queue settings (default)
MQQueue_openOptions. MQC.MQOO_BIND_ NOT_FIXED	MQ specific parameter
MQQueue_openOptions. MQC.MQOO_BIND_ON_ OPEN	MQ specific parameter
MQQueue.openOptions. MQC.MQOO_INPUT_AS_ Q_DEF	Accept queue settings for persistence etc. (Must always be set when reading) **Possible values:** • *yes*: Apply queue settings • *no*: Do not apply queue settings (default)
MQQueue.openOptions. MQC.MQOO_INPUT_SHARED	MQ specific parameter
MQQueue.openOptions. MQC.MQOO_INPUT_ EXCLUSIVE	MQ specific parameter

MQQueue.openOptions.MQC.MQOO_INQUIRE	MQ specific parameter
MQQueue.openOptions.MQC.MQOO_PASS_ALL_CONTEXT	MQ specific parameter
MQQueue.openOptions.MQC.MQOO_PASS_IDENTITY_CONTEXT	MQ specific parameter
MQQueue.openOptions.MQC.MQOO_SAVE_ALL_CONTEXT	MQ specific parameter
MQQueue.openOptions.MQC.MQOO_SET	MQ specific parameter
MQQueue.openOptions.MQC.MQOO_SET_ALL_CONTEXT	MQ specific parameter
MQQueue.openOptions.MQC.MQOO_SET_IDENTITY_CONTEXT	MQ specific parameter
MQPutMessageOptions.options.MQC.MQPMO_NEW_MSG_ID	MQ specific parameter
MQPutMessageOptions.options.MQC.MQPMO_SYNCPOINT	Use syncpoint when writing **Possible values:** • *yes*: Use syncpoint when writing • *no*: Do not use syncpoint when writing (default)
MQQueue.closeOptions.MQC.MQCOO_DELETE	MQ specific parameter
MQQueue.closeOptions.MQC.MQCOO_DELETE_PURGE	MQ specific parameter

MQMessage.format	Message format **Possible values:** • *MQC.MQFMT_STRING*: message consists entirely of characters • *MQC.MQFMT_ADMIN*: Command server request/response • *MQC.MQFMT_COMMAND1*: 1 command server response message type 1 • *MQC.MQFMT_COMMAND2*: 2 Command server response message type2 • *MQC.MQFMT_DEAD_LETTER_HEADER*: Dead-Letter-Header • *MQC.MQFMT_EVENT*: Event • *MQC.MQFMT_NONE*: No format name, no conversion takes place (i.e. the bytes are delivered original); if additionally parameter toXML is activated, then the message is Base64 encoded • *MQC.MQFMT_PCF*: User defined message in *Programmable Command Format* • *MQC.MQFMT_TRIGGER*: Trigger message • *MQC.MQFMT_XMIT_Q_HEADER*: Transmission Queue Header • *(no value)*: Do not convert (corresponds to MQC.MQFMT_NONE) (default)
MQMessage.applicationIdData	MQ specific parameter
MQMessage.applicationOriginData	MQ specific parameter
MQMessage.backoutCount	MQ specific parameter
MQMessage.messageFlags.MQC.MQMT_REQUEST	MQ specific parameter
MQMessage.replyToQueueName	Name of the ReplyToQueue for this message
MQMessage.replyToQueueManagerName	Name of the ReplyToQueueManager for this message **Possible values:** Any string
MQMessage.characterSet	Charset (CCSID) for the message; When reading: the expected charset; When writing: the charset that gets the message. **Possible values:** Any CCSID (e.g. *1208*)

MQMessage.correlationId	CorrelationID with 24 byte length (this is Base64 encoded as well as the returned MessageID). **Possible values:** Arbitrary string of 32 characters containing a Base64 encoded CorrelationID with 24 bytes length
MQMessage.encoding	Character encoding of the message
MQMessage.expiry	Time in tenths of seconds after which the message becomes invalid and is deleted from the queue. **Possible values:** Any integer
MQMessage.feedback	MQ specific parameter
MQMessage.groupId	MQ specific parameter
MQMessage.messageId	MQ specific parameter
MQMessage.messageSequenceNumber	MQ specific parameter
MQMessage.messageType	MQ specific parameter
MQMessage.offset	MQ specific parameter
MQMessage.originalLength	MQ specific parameter
MQMessage.persistence	MQ specific parameter
MQMessage.priority	MQ specific parameter
MQMessage.putApplicationType	MQ specific parameter
MQMessage.report	MQ specific parameter
MQMessage.userId	MQ specific parameter
MQGetMessageOptions.options.MQC.MQGMO_SYNCPOINT	Use syncpoint when receiving messages (required for commit and rollback operations). Possible values: • *yes*: Use syncpoint when receiving • *no*: Do not use syncpoint when receiving (default)

MQGetMessageOptions.options.MQC.MQGMO_WAIT	Wait during readout if there is no message in the queue
MQGetMessageOptions.options.MQC.MQGMO_NO_WAIT	When reading out, do not wait if there is no message in the queue and cancel immediately **Possible values:** • *yes*: Do not wait and cancel immediately • *no*: Wait (default)
MQGetMessageOptions.options.MQC.MQGMO_ACCEPT_TRUNCATED_MSG	MQ specific parameter
MQGetMessageOptions.options.MQC.MQGMO_BROWSE_FIRST	MQ specific parameter
MQGetMessageOptions.options.MQC.MQGMO_BROWSE_MSG_UNDER_CURSOR	MQ specific parameter
MQGetMessageOptions.options.MQC.MQGMO_BROWSE_NEXT	MQ specific parameter
MQGetMessageOptions.options.MQC.MQGMO_CONVERT	Enable automatic conversion to MQ **Possible values:** • *yes*: Enable conversion • *no*: No conversion (default)
MQGetMessageOptions.options.MQC.MQGMO_FAIL_IF_QUIESCING	MQ specific parameter
MQGetMessageOptions.options.MQC.MQGMO_LOCK	MQ specific parameter
MQGetMessageOptions.options.MQC.MQGMO_MSG_UNDER_CURSOR	MQ specific parameter

MQGetMessageOptions.options.MQC.MQGMO_NONE	Ignore all other *GetOptions* **Possible values:** • *yes*: Ignore all other GetOptions • *no*: Do not ignore other GetOptions (default)
MQGetMessageOptions.options.MQC.MQGMO_MARK_SKIP_BACKOUT	MQ specific parameter
MQGetMessageOptions.options.MQC.MQGMO_NO_SYNCPOINT	Deactivate Syncpoint **Possible values:** • *yes*: deactivate Syncpoint (no transaction is possible with it) • *no*: Do not deactivate Syncpoint (default)
MQGetMessageOptions.options.MQC.MQGMO_SYNCPOINT_IF_PERSISTENT	Use Syncpoint if the message is marked as persistent **Possible values:** • *yes*: Use syncpoint for persistent messages • *no*: Do not use syncpoint for persistent messages (default)
MQGetMessageOptions.options.MQC.MQGMO_UNLOCK	MQ specific parameter
MQGetMessageOptions.options.MQC.MQGMO_ALL_MSGS_AVAILABLE	MQ specific parameter
MQGetMessageOptions.options.MQC.MQGMO_ALL_SEGMENTS_AVAILABLE	MQ specific parameter
MQGetMessageOptions.options.MQC.MQGMO_COMPLETE_MSG	MQ specific parameter
MQGetMessageOptions.options.MQC.MQGMO_LOGICAL_ORDER	MQ specific parameter
MQGetMessageOptions.matchOptions.MQC.MQGMO_MATCH_CORREL_ID	MQ specific parameter

MQGetMessageOptions.matchOptions.MQC.MQGMO_MATCHGROUP_ID	MQ specific parameter
MQGetMessageOptions.matchOptions.MQC.MQGMO_MATCH_MSG_ID	MQ specific parameter
MQGetMessageOptions.matchOptions.MQC.MQGMO_MATCH_MSG_SEQ_NUMBER	MQ specific parameter
MQGetMessageOptions.matchOptions.MQC_MQGMO_NONE	MQ specific parameter
MQGetMessageOptions.waitInterval	Time in milliseconds to wait during readout if there are no messages in the queue (see parameter MQGetMessageOptions.options.MQC.MQGMO_WAIT). Message reception is not aborted if it was started during the wait interval. **Possible values:** Any positive integer
MQGetMessageOptions.resolvedQueueName	MQ specific parameterMQ specific parameter

For more information on MQ-specific parameters, visit https://www.ibm.com/docs/en/ibm-mq/8.0?topic=java-mqenvironment.

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

Example

The generated XML document when `Operation` property is set to *`Receive`* has the following structure:

```
<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>
```

For more information about the *MQ Series* specific parameters, please refer to its vendor *IBM*s documentation *WebSphere MQ - Using Java* in Chapter *9 The WebSphere MQ base Java classes and interfaces*.

10.14.4.3 Websphere MQ Connector

`Websphere MQ Connector`: Sends, receives and counts messages from an *IBM Websphere MQ* message queue, supporting *MQ* version 7.

Former name until version 4.5.1: *IBM Websphere MQ Adapter*

Note that this adapter works independently from the queue configuration of the *MQ Series Transfer Adapter* (within `x4.ear` this is done in the `conf/QueueConfig.xml` file).

Properties

Operation	Determines which operation is executed with the message queue. Possible values: • *`Send`*: Send data from the previous process step to the *MQ Series* queue. • *`Receive`*: Read message into the Technical Process • *`Commit`*: Complete transaction • *`Rollback`*: Cancel transaction and undo all changes • *`Count`*: Output the number of messages in the queue in an XML document.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.ibmmq.MQAdapter*: Main class (default)
toXML	If property Operation was set to *Receive*: Output the result of the process step as an XML document; if the `MQMessage.format` parameter is also set to *`NONE`*, the message is Base64-encoded. **Possible values:** • *`true`*: Generate XML document containing header and body of the message with all properties (see example). • *`false`*: Generate an MQ message as a result of the process step (default).
ccdtUrl	URL to a *Client Channel Definition Table (CCDT)* There are three different ways to define the `ccdtUrl`: 1. Via the Java arguments `-DMQCHLLIB` and `-DMQCHLTAB` when starting the *X4 Server* **Possible values:** `-DMQCHLLIB`:Path to the *table* file `-DMQCHLTAB`: *AMQCLCHL.TAB* (IBM default name) 2. Through the Java argument *`-Dx4.mqseries.ccdtUrl`* *Example:* `-Dx4.mqseries.ccdtUrl=file:///`*`Path_to_Table`*`/AMQCLCHL.TAB` 3. Via the `ccdtUrl` parameter in the adapter itself The above priority is applied.
RemoteQueueManager_name	Name of the Queue Manager
MQEnvironment_hostname	Host name of the addressed MQ server; applies Java VM-wide (the parameter is only required for local tests on development systems, otherwise it remains empty). **Possible values:** Any string (e.g. *MQSR2402*)
MQEnvironment_channel	Name of the channel; applies Java VM-wide (only needed for local tests on development systems, otherwise empty). **Possible values:** Any string (e.g. *MQSR2402.CLS25740.DT*)

MQEnvironment_CCSID	CCSID (Charset) of all nachichten for the entire Java VM; therefore, use with caution (should not be used!). **Possible values:** Any CCSID (e.g. *1208*)
MQEnvironment_userID	User ID for authentication of the *WebSphere MQ client*
MQEnvironment_ localAddress Setting	MQ specific parameter
MQEnvironment_ password	Password for authentication of the *WebSphere MQ Client*
MQEnvironment_port	Port of the MQ server; applies Java VM-wide (only needed for local tests on development systems, otherwise empty). **Possible values:** Any integer (e.g. *1414*)
MQEnvironment_ properties_MQC_ TRANSPORT_PROPERTY	Setting for the connection **Possible values:** *MQC.TRANSPORT_MQSERIES*: Only allowed value
MQEnvironment_ properties_MQC_ CCSID_PROPERTY	CCSID (Charset) for the connection **Possible values:** Any integer (e.g. *1208*)
MQEnvironment_ properties_MQC_ USER_ID_PROPERTY	User ID for the connection **Possible values:** Any string
MQEnvironment_ properties_MQC_ CHANNEL_PROPERTY	Name of the channel for the connection **Possible values:** Any string (e.g. *MQSR2402.CLS25740.DT*)
MQEnvironment_ properties_MQC_ CONNECT_OPTIONS_PROPERTY	Connection options **Possible values:** Any string
MQEnvironment_ properties_MQC_ HOST_NAME_PROPERTY	Host name of the addressed MQ server for the connection **Possible values:** Any string (e.g. *MQSR2402*)
MQEnvironment_ properties_MQC_ LOCAL__ADDRESS_PROPERTY	Local address for the connection **Possible values:** Any string
MQEnvironment_ properties_MQC_ ORB_PROPERTY	ORB property for the connection **Possible values:** Any string

MQEnvironment_ properties_MQC_ PASSWORD_PROPERTY	Password property for establishing the connection to authenticate the *WebSphere MQ Client* (corresponds to parameter MQEnvironment_password). **Possible values:** Any string
MQEnvironment_ properties_MQC_ PORT_PROPERTY	Port for the connection **Possible values:** Any string
MQEnvironment_ properties_MQC_ RECEIVE_EXIT_PROPERTY	*Receive Exit* property for the connection **Possible values:** Any string
MQEnvironment_ properties_MQC_ SECURITY_EXIT_PROPERTY	*Security Exit* property for the connection **Possible values:** Any string
MQEnvironment_ properties_MQC_ SEND_EXIT_PROPERTY	*Send Exit* property for the connection **Possible values:** Any string
MQQueueManager_name	Name of the QueueManager (only necessary for local tests on development systems, otherwise empty)
MQQueue_name	Queue name **Possible values:** Any valid string
MQQueue_dynamicQueue Name	Name of the *session queue* for this message (Should not be used in the future) **Possible values:** Any valid string
MQQueue_alternateUserId	Alternative user ID to check access to the queue when it is open; this parameter cannot be set while the object is open
MQQueue_openOptions_ MQC_MQOO_OUTPUT	(Must be set for write access)
MQQueue_openOptions_ MQC_MQOO_FAIL_IF_ QUIESCING	Allows the MQ Manager to disconnect the listener (e.g. at shutdown).
MQQueue_openOptions_ MQC_MQDL_SUPPORTED	MQ specific parameter
MQQueue_openOptions_ MQC_MQOO_ALTERNATE_ USER_AUTHORITY	MQ specific parameter

MQQueue_openOptions_MQC_MQOO_BIND_AS_QDEF	Apply queue settings for persistence etc. (should always be set) **Possible values:** • *true*: Apply queue settings • *false*: Do not apply queue settings (default)
MQQueue_openOptions_MQC_MQOO_BIND_NOT_FIXED	MQ specific parameter
MQQueue_openOptions_MQC_MQOO_BIND_ON_OPEN	MQ specific parameter
MQQueue_openOptions_MQC_MQOO_INPUT_AS_Q_DEF	Accept queue settings for persistence etc. (Must always be set when reading) **Possible values:** • *true*: Apply queue settings • *false*: Do not apply queue settings (default)
MQQueue_openOptions_MQC_MQOO_INPUT_SHARED	MQ specific parameter
MQQueue_openOptions_MQC_MQOO_INPUT_EXCLUSIVE	MQ specific parameter
MQQueue_openOptions_MQC_MQOO_INQUIRE	MQ specific parameter
MQQueue_openOptions_MQC_MQOO_PASS_ALL_CONTEXT	MQ specific parameter
MQQueue_openOptions_MQC_MQOO_PASS_IDENTITY_CONTEXT	MQ specific parameter
MQQueue_openOptions_MQC_MQOO_SAVE_ALL_CONTEXT	MQ specific parameter
MQQueue_openOptions_MQC_MQOO_SET	MQ specific parameter
MQQueue_openOptions_MQC_MQOO_SET_ALL_CONTEXT	MQ specific parameter
MQQueue_openOptions_MQC_MQOO_SET_IDENTITY_CONTEXT	MQ specific parameter
MQPutMessageOptions_options_MQC_MQPMO_NEW_MSG_ID	MQ specific parameter

MQPutMessageOptions_options_MQC_MQPMO_SYNCPOINT	Use Syncpoint when writing messages (required for the operations Commit and Rollback) **Possible values:** • *true*: Use Syncpoint when writing • *false:* Do not use syncpoint when writing (default)
MQQueue_closeOptions_MQC_MQCOO_DELETE	MQ specific parameter
MQQueue_closeOptions_MQC_MQCOO_DELETE_PURGE	MQ specific parameter
MQMessage_format	Message format **Possible values:** • *MQC.MQFMT_STRING*: message consists entirely of characters • *MQC.MQFMT_ADMIN*: Command server request/response • *MQC.MQFMT_COMMAND1*: 1 command server response message type 1 • *MQC.MQFMT_COMMAND2*: 2 Command server response message type2 • *MQC.MQFMT_DEAD_LETTER_HEADER*: Dead-letter Header • *MQC.MQFMT_EVENT*: Event • *MQC.MQFMT_NONE*: No format name, no conversion takes place (i.e. the bytes are delivered original); if additionally parameter toXML is activated, then the message is Base64 encoded • *MQC.MQFMT_PCF*: Custom message in the *Programmable Command Format* • *MQC.MQFMT_TRIGGER*: Trigger message • *MQC.MQFMT_XMIT_Q_HEADER*: Transmission Queue Header • *(no value)*: Do not convert (corresponds to MQC.MQFMT_NONE) (default)
MQMessage_applicationIdData	MQ specific parameter
MQMessage_applicationOriginData	MQ specific parameter
MQMessage_backoutCount	MQ specific parameter

MQMessage_messageFlags_MQC_MQMT_REQUEST	MQ specific parameter
MQMessage_replyToQueueName	Name of the ReplyToQueue for this message
MQMessage_replyToQueueManagerName	Name of the ReplyToQueueManager for this message **Possible values:** Any string
MQMessage_characterSet	Charset (CCSID) for the message; When reading: the expected charset; When writing: the charset that gets the message. **Possible values:** Any CCSID (e.g. *1208*)
MQMessage_correlationId	CorrelationID with 24 byte length (this is Base64 encoded as well as the returned MessageID). **Possible values:** Any string with 32 characters containing a Base64-encoded CorrelationID with 24 bytes length
MQMessage_encoding	Character encoding of the message
MQMessage_expiry	Time in tenths of a second after which the message becomes invalid and is deleted from the queue. **Possible values:** Any integer
MQMessage_feedback	MQ specific parameter
MQMessage_groupId	MQ specific parameter
MQMessage_message Id	MQ specific parameter
MQMessage_messageSequenceNumber	MQ specific parameter
MQMessage_messageType	MQ specific parameter
MQMessage_offset	MQ specific parameter
MQMessage_originalLength	MQ specific parameter
MQMessage_persistence	MQ specific parameter
MQMessage_priority	MQ specific parameter

MQMessage_put ApplicationType	MQ specific parameter
MQMessage_report	MQ specific parameter
MQMessage_userId	MQ specific parameter
MQGetMessageOptions_ options_MQC_MQGMO_ SYNCPOINT	Use syncpoint when receiving messages (required for commit and rollback operations). **Possible values:** • *true*: Use Syncpoint when receiving • *false*: Do not use syncpoint when receiving (default)
MQGetMessageOptions_ options_MQC_MQGMO_WAIT	Wait during readout if there is no message in the queue
MQGetMessageOptions_ options_MQC_MQGMO_ NO_WAIT	When reading out, do not wait if there is no message in the queue and cancel immediately **Possible values:** • *true*: Do not wait and abort immediately • *false*: Wait (default)
MQGetMessageOptions_ options_ MQC_MQGMO_ ACCEPT_TRUNCATED_MSG	MQ specific parameter
MQGetMessageOptions_ options_MQC_MQGMO_ BROWSE_FIRST	MQ specific parameter
MQGetMessageOptions_ options_MQC_MQGMO_ BROWSE_MSG_UNDER_CURSOR	MQ specific parameter
MQGetMessageOptions_ options_MQC_MQGMO_ BROWSE_NEXT	MQ specific parameter
MQGetMessageOptions_ options_MQC_MQGMO_ CONVERT	Enable automatic conversion to MQ **Possible values:** • *true*: Activate conversion • *false*: No conversion (default)
MQGetMessageOptions_ options_MQC_MQGMO_FAIL_ IF_QUIESCING	MQ specific parameter

MQGetMessageOptions_ options_MQC_MQGMO_LOCK	MQ specific parameter
MQGetMessageOptions_ options_MQC_MQGMO_ MSG_UNDER_CURSOR	MQ specific parameter
MQGetMessageOptions_ options_MQC_MQGMO_NONE	Ignore all other *GetOptions* **Possible values:** • `true`: Ignore all other GetOptions • `false`: Do not ignore other GetOptions (default)
MQGetMessageOptions_ options_MQC_MQGMO_ MARK_SKIP_BACKOUT	MQ specific parameter
MQGetMessageOptions_ options_MQC_ MQGMO_NO_SYNCPOINT	Deactivate Syncpoint **Possible values:** • `true`: deactivate Syncpoint (no transaction is possible with it) • `false`: Do not deactivate Syncpoint (default)
MQGetMessageOptions_ options_MQC_MQGMO_ SYNCPOINT_IF_PERSISTENT	Use Syncpoint if the message is marked as persistent **Possible values:** • `true`: Use syncpoint for persistent messages • `false`: Do not use syncpoint for persistent messages (default)
MQGetMessageOptions_ options_MQC_MQGMO_ UNLOCK	MQ specific parameter
MQGetMessageOptions_ options_MQC_MQGMO_ ALL_MSGS_AVAILABLE	MQ specific parameter
MQGetMessageOptions_ options_MQC_MQGMO_ALL_ SEGMENTS_AVAILABLE	MQ specific parameter
MQGetMessageOptions_ options_MQC_ MQGMO_COMPLETE_MSG	MQ specific parameter
MQGetMessageOptions_ options_MQC_ MQGMO_LOGICAL_ORDER	MQ specific parameter

MQGetMessageOptions_ matchOptions_MQC_ MQGMO_MATCH_CORREL_ID	MQ specific parameter
MQGetMessageOptions_ matchOptions_MQC_ MQGMO_MATCH_GROUP_ID	MQ specific parameter
MQGetMessageOptions_ matchOptions_MQC_ MQGMO_MATCH_MSG_ID	MQ specific parameter
MQGetMessageOptions_ matchOptions_MQC_ MQGMO_MATCH_MSG_ SEQ_NUMBER	MQ specific parameter
MQGetMessageOptions_ matchOptions_ MQC_MQGMO_NONE	MQ specific parameter
MQGetMessageOptions_ waitInterval	Time in milliseconds to wait during readout if there are no messages in the queue (see parameter `MQGetMessageOptions_options_MQC_MQGMO_WAIT`). Message reception is not aborted if it was started during the wait interval. **Possible values:** Any positive integer
MQGetMessageOptions_ resolvedQueueName	MQ specific parameter

Further information on the *MQ Series* specific parameters can be obtained from its producer *IBM* in the English language documentation *WebSphere MQ - Using Java*. In chapter *9 The WebSphere MQ base Java classes and interfaces.*

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

Operation: Receive

With Operation `Receive` messages can be read from the queue into the Technical Process.

Input

For Operation `Receive` the adapter does not expect an input document.

Output

For example, for Operation `Receive` and if Property `Operation` is set to *`Receive`*, the adapter outputs the following XML document:

```
<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
For Operation `Count` the adapter does not expect an input document.

Output
For Operation `Count`, the adapter outputs an XML document that contains the number of messages in the queue in the root `<Count>` element.

```
<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`: Reads a Microsoft Excel document (in Excel 97-2003 `.xls` format or in the XML-based OOXML format `.xlsx`) and outputs its content as an XML document.

Properties

Operation	Determines which operation the adapter executes. **Possible values:** • *`Read Spreadsheet`*: Read Excel document

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** • *en.softproject.integration.adapter.excel.ExcelReader*: Main class (default)
Formatter	Formatter used (has an influence on the XML output, see below) **Possible values:** • *de.softproject.integration.adapter.excel.DefaultFormatter*: Output each cell content of the Excel document in an element <Cell/> (default). • *de.softproject.integration.adapter.excel.NumberedCellFormatter*: Output each cell content of the Excel file in an XML element with cell number (e.g. *<Cell_0/>*). • *de.softproject.integration.adapter.excel.LabeledCellFormatter*: Output each cell content of the Excel file in an XML element with cell letters (e.g. *<Cell_A/>*)

`MinimumInflateRatio`	Optional. Specifies the minimum inflate ratio which is allowed when reading a file. ⓘ In Apache POI, a security check was introduced to intercept attacks by so-called ZIP bombs. In these attacks, a file multiplies its size when it is unpacked to memory. This can be problematic if the Java Virtual Machine's memory cannot handle the size of the decompressed file, which may lead to a server failure. The introduced security check uses a minimum decompression ratio. This value measures the difference between the amount of memory used by a file in the file system and the amount of memory required when reading the file. **Possible values:** • any integer between 0 and 100 **Please note:** • The default value is 1. With this value, the adapter fails if, for example, the file has a size of 1 KB in the file system, but a size of 100 KB in the memory. • A value of 0 deactivates the security check. • With a value of 100, the adapter does not read any of the provided documents, since a file is in no case smaller in memory than in the file system. • If the specified value is not in the allowed range (between 0 and 100) or if it is invalid, the default value is used.

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Output

The following XML structure is output with the standard formatter `de.softproject.integration.adapter.excel.DefaultFormatter`:

```
<Workbook>
   <Sheet name="Name des Tabellenblatts">
      <Row rowId="Zeilennummer">
         <Cell colId="Spalten-ID" type="Datentyp" format="Formatierung"
               formula="Formel"><!--Zelleninhalt--></Cell>
      </Row>
   </Sheet>
</Workbook>
```

The following XML structure is output with the formatter `de.softproject.integration.adapter.excelWriter.NumberedCellFormatter`:

```
<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>
```

The following XML structure is output with the formatter `de.softproject.integration.adapter.excel.LabeledCellFormatter`:

```
<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`: Converts an XML document with a specified table structure into a *Microsoft Excel* file (in `.xls` format *Excel 97 - 2003* or XML-based OOXML format `.xlsx`)or fills an existing Excel file (in `.xls` format *Excel 97 - 2003* or XML-based *OOXML* format `.xlsx`) with data from an XML document with a specified table structure.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • `Create Spreadsheet`: Output an Excel document with a spreadsheet that has the MIME type *`Application/Vnd.ms-excel`*. • `Fill Template`: Fill an existing Excel template, which is specified in the `TemplateURL` parameter and can have any number of spreadsheets, with data.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *en.softproject.integration.adapter.excel.ExcelWriter*: Main class (default)
generateOOXML	Determine Excel file format (Microsoft Excel 97 - 2003 or OOXML) **Possible values:** • *true*: Create XML-based Excel document (Office Open XML; .xlsx) • *false*: Create Excel document in Microsoft Excel 97 - 2003 (.xls) format (default)
TemplateUrl	URL to the Excel template (if the Operation property is set to *Fill Template*). Depending on the file format selected in parameter *generateOOXML*, the specified template must be in the same format; e.g. an `.xlsx` template must be used if parameter *generateOOXML* is set. The value of the parameter `TemplateUrl` can be overwritten by the attribute *templateUrl="..."* in the input XML (see below). **Possible values:** • URL with file:// protocol specification (e.g. *file:///%USER_COLLECTION%/Project/Template.xls*) • URL with xstore:// protocol specification (e.g. *xstore://Project/Folder/Template.xlsx*)
recalc	Recalculate formulas and update field values before saving the document **Possible values:** • *true*: Recalculate formulas • *false:* Leave values of fields with formulas unchanged (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

Depending on the selected operation, the adapter expects a specific input structure:

- **Operation `Create Spreadsheet`:**
  For this operation, the adapter expects the following XML structure, which corresponds to the structure of (X)HTML tables. Here, the `<table>` element creates a table sheet, each `<tr>` element creates a new table row and each `<td>` element creates a new cell or `<th>` a header cell. If several table sheets are to be created, the `<table>` elements must be enclosed by a root element with any name.

  > The table name which is specified using the `name` attribute in the `<table>` element is restricted to 31 characters.

  > The formatting of the individual cells can be controlled via various attributes. These are explained in the section **Possible Attributes**.

  Example input for the Create Spreadsheet operation

```
<?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>
```

  > With the attribute `forceString="true"`, the content is not interpreted as a number or date, but written directly into the Excel document.
  > *Example:* *`008.07300`* is written into the Excel document as the number *`8.073`* without the attribute `forceString="true"`, with the attribute `forceString="true"` it is written into the Excel document as the string *`008.07300`*.

- **Operation `Fill Template`:**
  For this operation, the adapter expects the following XML structure. In the optional attribute `templateUrl="... "` you can specify a valid URL to a template file, e.g. *`templateUrl="file:///C:/Template.xlsx"`*. This specification overwrites the value of the adapter parameter `TemplateURL`. Each `<cell>` element corresponds to a cell of the table. For row and, if applicable, column numbers, the count starts at 0.

  > The formatting of the individual cells can be controlled via various attributes. These are explained in the section **Possible Attributes**.

  Example input for 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>
```

Possible attributes

The formatting of the individual cells can be controlled via the following attributes:

Attribut	Description

formatNumber	Number format **Possible values:** • `number1`: Number 1 • `number2`: Number 2 • `number3`: Number 3 • `number4`: Number 4 • `currency1`: Currency 1 • `currency2`: Currency 2 • `currency3`: Currency 3 • `currency4`: Currency 4 • `accounting`: Accounting • `date`: Date • `time`: Time • `percentage`: Percentage • `Fraktion`: Fracture • `scientific`: Science • `text`: Text • `postalCode`: Postcode • `telephone`: Telephone number If required, user-defined number formats can also be stored
fontName	Font, e.g. `Calibri`
horizontalAlignment	Horizontal alignment of the text within the cell **Possible values:** • `CENTER` • `CENTER_SELECTION` • `DISTRIBUTED` • `FILL` • `GENERAL` • `JUSTIFY` • `LEFT` • `RIGHT`
verticalAlignment	Vertical alignment of the text within the cell **Possible values:** • `CENTER` • `BOTTOM` • `TOP` • `DISTRIBUTED` • `JUSTIFY`
fontHeight	Font Size

fontColor	Font color **Possible values:** • CSS colour specifications, e.g. `Red`, `Blue`, `Green` etc. • RGB colour code, e.g. `240, 80, 0` • HEX colour code, e.g. `#124361`
bold	Bold font **Possible values:** • `true`: bold font • `false`: normal font
italic	Italic font **Possible values:** • `true`: italic font • `false`: normal font
underline	Type of underline **Possible values:** • `NONE` • `DOUBLE` • `DOUBLE_ACCOUNTING` • `SINGLE` • `SINGLE_ACCOUNTING`
cellColor	Fill colour of the cell **Possible values:** • CSS colour specifications, e.g. `Red`, `Blue`, `Green` etc. • RGB colour code, e.g. `240, 80, 0` • HEX colour code, e.g. `#124361`

border **borderLeft** **borderRight** **borderBottom**	Type of cell frame **Possible values:** • `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**	Cell frame colour **Possible values:** • Colour name, e.g. `Red`, `Blue`, `Green` etc. • RGB colour code, e.g. `240, 80, 0` • HEX colour code, e.g. `#124361`
orientation	Text alignment **Possible values:** • Number between *`90`* and *`-90`* • `HORIZONTAL` • `VERTICAL`
wrapText	*Optional attribute*: Allow automatic line break in cells **Possible values:** • `true`: Enable line break • `false`: Deactivate line break

autofit *Optional attribute*: Automatically adjust column width

Possible values:

- `true`: Activate automatic column width
- `false`: Disable automatic column width

Set the attribute

Please note:

- For the `Create Spreadsheet` operation, the attribute must be set within the first line in the `<td>` or `<th>` element.
- For the `Fill Template` operation, a new `columnProperties` element must first be inserted, in which the configuration of the individual columns - each in a `column` node - is carried out.

`columnProperties`	Groups the columns of a sheet. **Possible values:** `sheet`: Name of the Excel sheet for which the configuration is made.
`column`	Column within a sheet **Possible values:** `id`: Index of the column. The first index is `0`. If a negative value is given, an error message is displayed.

```
<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 *Optional attribute*: Define fixed column width

> If the `autofit` attribute is set, the `width` attribute is ignored.

Possible values: Number between *0* and *255.* If the specified number is greater than `255`, `255` is set as the maximum column width.

Set the attribute

Please note:

- For the `Create Spreadsheet` operation, the attribute must be set within the first line in the `<td>` or `<th>` element.
- For the `Fill Template` operation, a new `columnProperties` element must first be inserted, in which the configuration of the individual columns - each in a `column` node - is done.

`columnProperties`	Groups the columns of a sheet. **Possible values:** `sheet`: Name of the Excel sheet for which the configuration is made.
`column`	Column within a sheet **Possible values:** `id`: Index of the column. The first index is `0`. If a negative value is given, an error message is displayed.

```
<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 *Optional attribute*: Define row height

> Excel applies the automatic adjustment of the height with the `autofit` attribute only if a fixed row height was previously defined via the `height` attribute. If the `autofit` attribute is set for the row, the `heigth` attribute is ignored. In addition, the automatic line height is also controlled via the attribute `wrapText`.

Possible values: number between *0* and *409*. If the specified number is greater than `409`, `409` is set as the maximum line height.

Set the attribute

Please note:

- For the `Create Spreadsheet` operation, the attribute must be set within the `<tr>` element.
- For the `Fill Template` operation, a new `rowProperties` element must first be inserted in which the configuration of the individual rows - each in a `row` node - is carried out.

`rowProperties`	Groups the rows of a sheet. **Possible values:** `sheet`: Name of the Excel sheet for which the configuration is made.
`row`	Line within a sheet **Possible values:** id: Index of the line. The first index is `0`. If a negative value is given, an error message is displayed.

```
<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`: Reads metadata (document properties like author, last saved) of a *Microsoft Word* document (`.doc` or `.docx`) and outputs the data in a UTF-8-encoded XML document.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** *`Get`*: Read document properties from an *MS Word* document

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.adapter.wordheaderadapter.wordHeaderAdapter`*: Main class (Default)

Status values

1 (successful)	The adapter was executed successfully and the read document properties were outputted as XML document
0	The processed document has a wrong file format
-1(error)	The processed document is not a valid `.doc` or `.docx` file or other error occurred (see server protocol)

Input

The adapter expects a Microsoft Word document with the ending `.doc` or `.docx` as input.

Output

The adapter ouputs an XML document with the following structure:

```
<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 edited in word -->
        <CreatedOn>2017-12-07T08:22:00Z</CreatedOn>
        <Title>Titelproperty</Titel>
        <Subject>Subjectproperty</Subject>
        <Keywords>Keyword</Keywords>
        <Comments>Comment</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>
```

Explanation: The element `UserProperties` is only filled for files with the ending `.doc`.

> **Data format changes**
> Please note that the data format has changed. The formerly German format `dd.mm.yyyy hh:mm:ss` is now formatted according to ISO-8601 in the UTC timezone `yyyy-mm-ddThh:mm:ssZ`.

10.15.3 Microsoft Word to Text Converter

? Unknown Attachment `Microsoft Word to Text Converter`: Reads a *Microsoft Word* document (`.doc` in *Office 97-2003* format), extracts its text contents and outputs a text document. During conversion, all Word-specific formatting parameters are removed so that the pure content is available for further processing.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Convert`*: Extract text from an MS Word document.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.Word2XML`*: Main class (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.15.4 Working Time Calculator

`Working Time Calculator`: Calculates the working time.

> The *Working Time Calculator* replaces with version 6.0 of the X4 BPMS the outdated working time calculation functions of the Java class `de.softproject.xsl.WorkingTimeCalculator`, which were provided within XSL mappings.

Properties

`Operationen`	Defines the operation executed by the adapter **Possible values:** • `AddWorkingTimeMinutes`: Adds a certain number of working minutes to a start date and displays the end date and time • `GetWorkingTimeMinutes`: Calculates the working time between a start date and an end date and displays the working minutes • `GetEaster`: Calculates the Easter Sunday based on a year number

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** *de.softproject.x4.adapter.workingTimeCalculator*: Main class (default)

Status values

`1`	The operation was executed successfully and the output contains an XML document with the corresponding result

`-1`	An error occurred during the adapter's execution: • The required date format (`yyyy-MM-dd` bzw. `yyyy-MM-dd hh:mm`) was not followed. • The days of the week were not entered correctly. A correct German or English weekday is required. • No Input available.

Input

The adapter expects a predefined input XML structure for the various operations:

- Holidays or vacation days (non-working days) are defined in the date format `yyyy-MM-dd` within any number of `<Holiday>` elements. These are created as child elements of `<Holidays>`.
- Workdays (weekdays) are defined within `<WorkingDay>` elements. These are created as child elements of `<WorkingDays>`. The value of `<WorkingDay>` is the respective weekday in German or English, e.g. *`Montag`* oder *`Friday`*. The upper and lower case is not important.
- Special Days are defined in the date format `yyyy-MM-dd HH:mm-HH:mm` within any number of `<SpecialDay>` elements. These are created as child elements of `<SpecialDays>`.

  > `SpecialDays` are considered a special working day regardless of if it is a normal working day or a day off. Only the time defined within `<SpecialDay>` counts as working time.

> The working time calculation functions `dateDiff` or `dateDiffWorkingMinutes` of the obsolete Java class `de.softproject.xsl.WorkingTimeCalculator` have been replaced by the following functions:
>
> - `dateDiffWorkingMinutes` can be implemented by operation `GetWorkingTimeMinutes`
> - `dateDiff` can be implemented by defining all weekdays as `WorkingDay` and setting working hours from `00:00` to `23:59`. Neither `Holidays` nor `SpecialDays` must be defined.

- Operation `AddWorkingTimeMinutes`:

Sample input for the 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`:

Sample input for the 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`:

Sample input for the operation GetEaster

```
<WorkingTimeCalculator>
    <GetEaster>2019</GetEaster>
</WorkingTimeCalculator>
```

Output

The adapter outputs an XML document depending on the selected operation:

- Operation `AddWorkingTimeMinutes`:

Sample output for the operation AddWorkingTimeMinutes

```
<WorkingTimeCalculator>
    <addWorkingTimeMinutes>2019-12-24T21:50:00</addWorkingTimeMinutes>
</WorkingTimeCalculator>
```

- Operation `GetWorkingTimeMinutes`:

Sample output for the operation GetWorkingTimeMinutes

```
<WorkingTimeCalculator>
    <getWorkingTimeMinutes>2010.0</getWorkingTimeMinutes>
</WorkingTimeCalculator>
```

- Operation `GetEaster`:

Sample output for the 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`: Creates barcodes and adds them to an existing non-protected PDF document. The barcode graphics can be positioned, scaled and rotated on any PDF page. The barcode types *`Code 39`*, *`Code 128`*, *`Codabar`*, *`EAN 13`*, *`Code 25-Interleaved 2 of 5`* and *`POSTNET`* are supported.
The barcode data is stored in an adapter-specific input XML document. The adapter outputs a PDF document that has been supplemented with the specified barcodes.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`AddBarcodes`*: Add barcode(s) to the PDF document

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *en.softproject.integration.adapter.pdf.PDFBarcode*: Main class (default)
ReadPartially	Load PDF document into memory in multiple parts (can improve the performance) **Possible values:** • *yes*: Load in several parts one after the other into the working memory • *no*: Load PDF document in one piece into the working memory (default)
toXML	Output PDF document Base64-encoded in an XML document **Possible values:** • *yes*: Output XML document • *no*: Output PDF document as binary data stream (default).

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

Input

This adapter expects an adapter-specific input XML structure in which all data of each barcode graphic is stored (see below). If a PDF from the repository is to be used as the source document, specify its `xstore` URL in the `url` attribute in element `<PDF>`. Alternatively, if the source document is available as a base64-encoded string, you can store this string as the content of the element `<PDF encoding="base64">` (see commented out line).

```
<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>
```

Attributes in the input XML document

The *PDF Barcode Adder* expects an adapter-specific input XML structure according to the above pattern. The following attributes can be set:

url	xstore path specification to the input PDF document in the repository (e.g. *xstore://Project/Data/InputData.xml;* required).
height	Height of barcode graphic in *cm* or *pt* (required)
width	Width of the barcode graphic in *cm* or *pt* (required)
rad	Rotation of the barcode graphic in radians
degree	Rotation of the barcode graphic in degrees (required)
xPos	Distance of barcode graphic from left margin in *cm* or *pt* (required)
yPos	Distance of the barcode graphic from the bottom of the page in *cm* or *pt* (required)
type	Barcode type (required) Possible values: • *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	Page number in the PDF document on which the barcode graphic will be positioned (required).

Example

The following input XML document contains the data to create an *EAN 13* barcode graphic with 3 cm width and 20° slope on the first page of a PDF document in the 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`: Creates a PDF document with or without form fields from FDF/XFDF form data, extracts an FDF file from a PDF form document, sets form field values or converts a PDF form document into a flat PDF document.
Former name until version 4.5.1: *PDF Form Processing Adapter*

> The FDF/XFDF form data are saved in a text or XML document. This form data document must contain an absolute path to a PDF form document that serves as a template for the PDF document to be created (not relevant for the operations `Export FDF` and `Flatten`).

Properties

Operation	Determines which operation the adapter executes **Possible values:** • `Flatten XFDF`: Write XFDF data to a PDF form document (convert form fields to text). • `Flatten FDF`: Write FDF data into a PDF form document (convert form fields into text) • `Export FDF`: Extract FDF data from a PDF form document • `Import XFDF`: Write XFDF data into a PDF form document (form fields retained) • `Import FDF`: Write FDF data into a PDF form document (form fields retained) • `Set Fields`: Set form field values and, if necessary, convert the form fields to text in the output PDF document. • `Flatten`: Convert PDF form document to a flat PDF document (convert form fields to text).

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.pdf.PDFForm`*: Main class (default)
`PreserveExtendedAttributes`	Preserve the extended attributes in the PDF document when using the `Set Fields` operation, if it was defined via `flatten="false"` for a form field in the input XML that form fields should remain editable. **Possible values:** • *`yes`*: Receive the extended attributes after filling the form fields (default) • *`no`*: Extended attributes for the form fields in the PDF document are not preserved (this corresponds to the behaviour in previous versions of this adapter and is only relevant in case of compatibility problems).

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Operation: Flatten XFDF

The `Flatten XFDF` operation writes XFDF data (ISO 19444-1) to a PDF form document. The form fields are converted to text and cannot be edited.

Input

> The structure of an XFDF document is described in ISO 19444-1.

For `Flatten XFDF` Operation the adapter expects an XFDF document which specifies the absolute path to the associated PDF form in `<f>` element.

```
<?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

The output is a copy of the PDF form document referenced in the input with flat text.

For further information please visit the section

- Operation: Flatten XFDF
- Operation: Flatten FDF
- Operation: Export FDF
- Operation: Import XFDF
- Operation: Import FDF
- Operation: Set Fields
- Operation: Flatten

Operation: Flatten FDF

The `Flatten FDF` operation writes FDF data (ISO 19444-1) to a PDF form document. The form fields are converted to text and cannot be edited.

Input

For `Flatten FDF` operation the adapter expects an FDF document defining the absolute path to the associated PDF form in the `/F` parameter.

```
%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

The output is a copy of the PDF form document referenced in the input with flat text.

For further information please visit the section

- Operation: Flatten XFDF
- Operation: Flatten FDF
- Operation: Export FDF
- Operation: Import XFDF
- Operation: Import FDF
- Operation: Set Fields
- Operation: Flatten

Operation: Export FDF

The `Export FDF` operation extracts FDF data (ISO 19444-1) from a PDF form document.

Input

For `Export FDF` operation the adapter expects a PDF form document.

Output

The FDF data contained in the PDF form document will be output.

For further information please visit the section

- Operation: Flatten XFDF
- Operation: Flatten FDF
- Operation: Export FDF
- Operation: Import XFDF
- Operation: Import FDF
- Operation: Set Fields
- Operation: Flatten

Operation: Import XFDF

The `Import XFDF` operation writes XFDF data (ISO 19444-1) to a PDF form document. The form fields are preserved and can still be edited.

Input

> The structure of an XFDF document is described in ISO 19444-1.

For `Import XFDF` operation the adapter expects an XFDF document which defines the absolute path to the associated PDF form in `<f>` element.

```
<?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

The output is a copy of the PDF form document referenced in the input with filled form fields.

> For further information please visit the section
>
> - Operation: Flatten XFDF
> - Operation: Flatten FDF
> - Operation: Export FDF
> - Operation: Import XFDF
> - Operation: Import FDF
> - Operation: Set Fields
> - Operation: Flatten

Operation: Import FDF

The `Import FDF` operation writes FDF data (ISO 19444-1) to a PDF form document. The form fields are preserved and can still be edited.

Input

For `Import FDF` operation the adapter expects an FDF document defining the absolute path to the associated PDF form in the `/F` parameter.

```
%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

The output is a copy of the PDF form document referenced in the input with filled form fields.

For further information please visit the section

- Operation: Flatten XFDF
- Operation: Flatten FDF
- Operation: Export FDF
- Operation: Import XFDF
- Operation: Import FDF
- Operation: Set Fields
- Operation: Flatten

Operation: Set Fields

The `Set Fields` operation sets form field values in the output PDF document. Form fields are converted to text in the output PDF document and are no longer editable if the `flatten="true"` attribute has been declared.

Input

For Operation `Set Fields` the adapter expects an adapter-specific XML document which defines the absolute path to the associated PDF form in `<Document>` element and the form field data to be set in any number of `<Field>` elements. With attribute `flatten="true"` the form fields are converted to flat text in the generated PDF document, with `flatten="false"` the form fields remain editable.

```
<?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

The output is a copy of the PDF form document referenced in the input with flat text. The text cannot be edited because `flatten="true"` was defined in the input.

For further information please visit the section

- Operation: Flatten XFDF
- Operation: Flatten FDF
- Operation: Export FDF
- Operation: Import XFDF
- Operation: Import FDF
- Operation: Set Fields
- Operation: Flatten

Operation: Flatten

The *`Flatten`* operation converts a PDF form document into a PDF document. The form fields are converted to text and cannot be edited.

Input

For Operation `Flatten`, the adapter expects a PDF form document.

Output

A PDF document is generated as output.

For further information please visit the section

- 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`: Reads metadata structures from PDF documents or writes metadata structures to PDF documents.

Properties

`Operation`	Determines which operation the adapter executes **Possible values:** • `GetMetadata`: Read metadata • `SetMetadata`: Add/replace metadata

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** `en.softproject.integration.adapter.pdf.PDFMetadata`Main class (default)
`ReadPartially`	Load PDF document into memory in multiple parts (can improve the performance) **Possible values:** • *yes*: Load in several parts one after the other into the working memory • *no*: Load PDF document(s) into memory in one piece (default)
`toXML`	Output PDF document Base64-encoded in an XML document **Possible values:** • *yes*: Output XML document • *no*: Output PDF document as binary data stream (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

For more information, see the section

- Operation: GetMetadata
- Operation: SetMetadata

Operation: GetMetadata

The `GetMetadata` operation reads the metadata of a PDF file.

Input

PDF in PDF format

The following input XML document contains the path to a PDF document in the repository.

```
<?xml version="1.0" encoding="UTF-8"?>
<PDF url="xstore://Project/Folder/Input.pdf"/>
```

PDF in Base64 format

The following input XML document contains the Base64 string.

```
<?xml version="1.0" encoding="UTF-8"?>
<PDF encoding="base64">
    Base64-Zeichenkette
</PDF>
```

Output

If the `ToXML` parameter is **disabled**, the output looks like follows:

Output at parameter ToXML: yes

```
{CreationDate=D:20210610111922+02'00', ModDate=D:20210610111922+02'00'}
```

If the `ToXML` parameter is **enabled**, the output looks like follows:

Output at parameter ToXML: yes

```
<?xml version="1.0" encoding="UTF-8"?>
<Metadata>
   <CreationDate>D:20210610111922+02'00'</CreationDate>
   <ModDate>D:20210610111922+02'00'</ModDate>
</Metadata>
```

For further information please visit the section

- Operation: GetMetadata
- Operation: SetMetadata

Operation: SetMetadata

The `SetMetadata` operation adds metadata to a PDF file or replaces existing metadata.

PDF default metadata

Element	Description
`<Author>`	Author of the document
`<CreationDate>`	the creation date and time
`<Creator>`	The original application that created the document
`<Producer>`	The product that created the document.
`<Subject>`	Subject of the document
`<Title>`	Title of the document
`<Keywords>`	Keywords Can be separated by commas
`<ModDate>`	Date and time of the last change

You add custom metadata with a custom element. In the following example, a `Region` metadata is added with the value `Europe`:

```
<Region>Europe</Region>
```

Input

PDF in PDF format

The following input XML document contains the path to a PDF document in the repository and the metadata to add.

```
<?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 in Base64 format

The following input XML document contains the Base64 string and the metadata to be added.

```
<?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

If the `ToXML` parameter is **disabled**, the output can be saved directly as a buffer file in PDF format.

If the `ToXML` parameter is **enabled**, the output looks like follows:

Output at parameter ToXML: yes

```
<?xml version="1.0" encoding="UTF-8"?>
<PDF encoding="base64">
  Base64-Zeichenkette
</PDF>
```

For further information please visit the section

- Operation: GetMetadata
- Operation: SetMetadata

10.16.1.4 PDF to Text Converter

? Unknown Attachment `PDF to Text Converter`: Reads a PDF document (also password-protected), extracts the text contents of a specific page range or of the entire document and outputs the extracted character strings in an XML or text document with a freely selectable character encoding.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Extract`*: Extract text from the input PDF document.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.pdf.PDF2Text`*: Main class (default)
password	Password (for a protected PDF document) **Possible values:** Any string
startPage	First page number from which the texts are to be extracted **Possible values:** • Any positive integer or *`0`* • *`0`*: Start from the first page (default)
endPage	Last page number up to which the text extraction is to be carried out **Possible values:** • Any integer or *`0`* • *`0`*: Extract text to the last page (default)
encoding	Character encoding of the result document **Possible values:** Any valid character encoding (e.g. *`UTF-8`*).
force	Try to extract text even on invalid PDF pages **Possible values:** • *`yes`*: Process invalid PDF pages • *`no`*: Ignore invalid PDF pages (default)

toXML	Output text contents in an XML document **Possible values:** • *yes*: Output XML document • *no*: Output text document (default)

Status values

`-1`	The operation was executed successfully.
`1`	The operation failed due to a technical error.

10.16.2 PDF Image Processing

10.16.2.1 PDF Image Adder

`PDF Image Adder`: Adds one or more graphics to an existing non-protected PDF document or creates a new PDF document with the specified graphics. These can be specified both as a URL and as a Base64-encoded string.

Supported graphic formats are BMP, GIF, JPEG/JPG, PNG and TIFF. The graphics can be positioned, scaled and rotated on each PDF page. The information about the graphics is stored in an adapter-specific input XML document. The adapter outputs either a new PDF document or a PDF document to which the specified graphics have been added.

> As of adapter version 2.0.0, the EPS and WMF graphic formats are no longer supported.

If the input document does not contain a PDF element, a PDF document in DIN A4 format with the defined number of pages is first created before the graphics are added.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`AddImages`*: Add images to the PDF document

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *en.softproject.integration.adapter.pdf.PDFImage*: Main class (default)

ReadPartially	Load PDF document into memory in multiple parts (can improve the performance) **Possible values:** • *yes*: Load in several parts one after the other into the working memory • *no*: Load PDF document in one piece into the working memory (default)
toXML	Output PDF document Base64-encoded in an XML document **Possible values:** • *yes*: Output XML document • *no*: Output PDF document as binary data stream (default).

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

This adapter expects an adapter-specific input XML structure in which information about each graphic is stored. This includes width and height (in *cm* or *pt*), rotation, x- and y-position (distances from the lower left page margin in *cm* or *pt*), the URL to the graphic as well as the page on which the graphic is to be positioned. The input PDF document can be specified either as a URL or as a Base64-encoded string.

```
<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>
```

The following input XML document contains information for a PNG image file to be inserted on the first page of an existing PDF document (specified as URL or Base64 string).

```
<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>
```

The following input XML document contains information for three graphics that are to be merged into a new PDF document.

```
<?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`: Adds one or more images from *multipage Tagged Image File Format (TIFF)* files to an existing unprotected PDF document.

The images can be positioned, scaled and rotated on any PDF page. The information about the images is stored in an adapter-specific input XML document. The adapter outputs a PDF document to which the specified images have been added, as a binary data stream or as an XML document.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *AddImages*: Add images from multi-page TIFF files to the PDF document.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.pdf.PDFTiffImage*: Main class (default)
ReadPartially	Load PDF document into memory in multiple parts (can improve the performance) **Possible values:** • *yes*: Load in several parts one after the other into the working memory • *no*: Load PDF document(s) into memory in one piece (default)
toXML	Output PDF document Base64-encoded in an XML document **Possible values:** • *yes*: Output XML document • *no*: Output PDF document as binary data stream (default).

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

This function adapter expects an adapter-specific input XML structure in which information about each image is stored. This includes width and height (in *cm* or *pt*), rotation, x and y position (distances from the lower left page margin in *cm* or *pt*), the page in the multipage TIFF, the URL to the image, and the page in the PDF document where the image is to be positioned.

Each `<TIFF/>` element references a TIFF file and each `<image/>` element places an image in the PDF. If an input PDF document is a Base64 encoded data, the commented out code must be adjusted and commented in instead.

```
<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>
```

Example

The following input XML document contains information about two pages of a multi-page TIF image to be inserted into a new PDF document.

```
<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`: Converts and scales the first page of an existing, unprotected PDF document into an image format.

The following formats can be created: BMP, GIF, JPEG, JPEG 2000, PNG, PNM and TIFF. The adapter outputs the graphic in the desired file format as a binary data stream.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *Transform*: Convert PDF to image format

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.pdf.PDF2Image`*: Main class (default)
width	Width of the output image in pixels (ignored if parameter maxSize is set) **Possible values:** • Any positive integer, e.g. *`800`* • *`-1`* (in combination with a positive pixel number in parameter height): retain the original aspect ratio when scaling the image; in combination with the value -1 in parameter height the original image dimensions are retained
height	Height of the output image in pixels (ignored if parameter maxSize is set) **Possible values:** • Any positive integer, e.g. *`600`* • *`-1`* (in combination with a positive pixel number in parameter `width`): retain the original aspect ratio when scaling the image; in combination with the value `-1` in parameter `width` the original image dimensions are retained
page	Page within the PDF document that is to be output as an image. **Possible values:** • Any integer
maxSize	Maximum width or height of the output image in pixels (parameters `width` and `height` are ignored). **Possible values:** • Any integer • *`0`*: Use `width` and `height` parameters for scaling (default)

scaleAlgorithm	Scaling algorithm used **Possible values:** • *SCALE_DEFAULT*: Use standard algorithm • *SCALE_FAST*: Use fast algorithm • *SCALE_SMOOTH*: Use edge smoothing algorithm • *SCALE_REPLICATE*: Use pixel repetition algorithm • *SCALE_AREA-AVERAGING*: Use area averaging algorithm
outputFormat	Output image format **Possible values:** • *BMP*: Create Windows Bitmap Graphic • *GIF*: Create GIF compressed image • *JPEG*: Create JPEG-compressed image • *JPEG 2000:* Create image in *JPEG 2000* format • *PNG*: Create PNG image • *PNM*: Create image in *Portable Anymap* format • *TIFF*: Create image in *Tagged Image File Format*

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.16.2.4 PDF to TIFF Converter

? Unknown Attachment `PDF to TIFF Converter`: Converts a PDF document into a TIFF graphic (*Tagged Image File Format*) and outputs the graphic as a binary data stream. Multi-page PDF documents are converted into a multi-page TIFF graphic (*Multipage TIFF*).

Properties

Operation	Determines which operation the adapter executes **Possible values:** *Transformation*: Convert PDF document into a TIFF graphic.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.pdf2tiff.PDFToTiffAdapter`*: Main class (default)
compression	Compression type **Possible values:** • *`Deflate (1)`*: Lossless compression via deflate • *`JPEG (2)`*: Use JPEG compression (lossy) • *`None (3)`*: Do not compress (default) • *`Packbits (4)`*: Lossless compression via PackBits run-length encoding
deflateLvl	Deflate level (if *Deflate* compression type is selected and *Deflate* compression is to be applied to the entire TIFF file). **Possible values:** • Any integer from *`1`*–*`9`* (compresses fastest at *`1`*, most verbose at *`9`*). • *`0`*: Do not apply *deflate* compression to the entire TIFF image (default).
quality	For JPEG compression: Quality level of the (lossy) JPEG compression. **Possible values:** • Any integer from *`0`*–*`100`* (at *`0`* lowest quality, at *`100`* maximum quality) • *`0`*: Lowest quality, i.e. maximum compression (standard)
resX	Horizontal screen/print resolution value **Possible values:** • Any integer (e.g. *`72`*) • *`0`*: Do not set a resolution value

resY	Vertical screen/print resolution value **Possible values:** • Any integer (e.g. *72*) • *0*: Do not set a resolution value
resUnit	Unit for screen/print resolution **Possible values:** • *No absolute unit*: Do not set an absolute unit • *Inch*: Use inch as unit (DPI or PPI) • *Centimeter*: Use centimetre as unit

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

An (optional) CPU type/operating system-dependent *JAI MediaLib* library is not supplied. The adapter therefore shows the following error message in the server log, but this is not relevant for adapter execution:

```
ERROR [STDERR] Could not find mediaLib accelerator wrapper classes.
 Continuing in pure Java mode."
```

To switch off the *JAI MediaLib* library, set the following Java VM option (e.g. in the start script):

```
@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`: Creates a PDF document in 210 x 297 mm (DIN A4) format from a TIFF file (*Multipage Tagged Image File Format*), where each page in the TIFF file corresponds to one page in the PDF document. The graphics can be positioned in the PDF document and scaled proportionally based on the width of the image.

Notice that `.tif` files with alpha channels are not supported.

Properties

`Operation`	Determines which operation the adapter executes **Possible values:** `Convert`: Create PDF document from a multi-page TIFF file.

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.pdf.Tiff2Pdf*: Main class (default)
`toXML`	Output PDF document Base64-encoded in an XML document **Possible values:** • `yes`: Output XML document • `no`: Output PDF document as binary data stream (default).
`Header`	Text that appears as a heading in the PDF on each page **Possible values:** Any string
`xPos`	Distance from the left side of the page in pt **Possible values:** Any integer
`yPos`	Distance from the bottom of the page in pt **Possible values:** Any integer
`scale`	Scale images and fit them into the PDF page **Possible values:** • `yes`: Scale images • `no`: Do not scale images, i.e. use the resolution specified in the TIFF file (default).

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Output

The adapter outputs a PDF document as a binary data stream or as an XML document containing all pages of the multi-page TIFF file.

10.16.3 PDF Split and Merge

10.16.3.1 PDF Merger

`PDF Merger`: Merges multiple unprotected PDF documents into one PDF document.

This adapter expects an adapter-specific input XML structure which contains multiple Base64-encoded strings with the data of PDF documents. The adapter outputs an XML document which contains a Base64-encoded string with the data of the merged PDF document.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Merge`*: Merge multiple PDF documents into one PDF document.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.PDFMerger`*: Main class (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

This function adapter expects an adapter-specific input XML structure in which all PDF document data is stored as Base64-encoded strings in the desired order (defined in the `order` attribute).

```
<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>
```

Output

The successfully executed adapter returns an XML document with the following pattern:

```
<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`: Determines the number of pages in a PDF document that is being processed as an input document, and outputs an XML document with the page number.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`GetPageCount`*: Determine number of pages in PDF (default)

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.pdf.PDFPageCount`*: Main class (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Output

The function adapter outputs an XML document with the following structure:

```
<Pages><!--Anzahl der Seiten--></Pages>
```

10.16.3.3 PDF Recombiner

`PDF Recombiner`: Merges pages from multiple unprotected PDF documents into a new PDF document.

This adapter expects an adapter-specific input XML structure which contains all processing information for the PDF documents. The adapter outputs the data of a merged PDF document as a binary data stream or as an XML document.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`RecombinePDF`*: Merge pages from different PDF documents into a new PDF document

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.pdf.PDFRecombiner`*: Main class (default)
ReadPartially	Load PDF document into memory in multiple parts (can improve the performance) **Possible values:** • *`yes`*: Load in several parts one after the other into the working memory • *`no`*: Load PDF document(s) into memory in one piece (default)
toXML	Output PDF document Base64-encoded in an XML document **Possible values:** • *`yes`*: Output XML document • *`no`*: Output PDF document as binary data stream (default).

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

This adapter expects an adapter-specific input XML structure which contains all processing information for the generated PDF document.

```
<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>
```

Explanation:

- `<PDF/>`: references one input document.
  If an input PDF document is base64 encoded data, the commented out code must be adjusted and commented in instead.
- `<Range/>` defines a page range from start to end page.
- `<Page/>` defines a single page.
- `<All/>` defines all pages of an input PDF document.
- `<Empty/>` defines blank pages in the generated PDF document.

Example

The following input XML document contains information about two PDF documents that are to be merged into a new PDF document with the defined pages and a total of three blank pages.

```
<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`: Splits an unprotected PDF document into several new PDF documents.

This adapter expects an adapter-specific input XML structure which contains the path to the processed PDF document in the repository. The adapter outputs the data of a PDF part document as a binary data stream or as an XML document per iteration.

Properties

`Operation`	Determines which operation the adapter executes **Possible values:** • `Split Iterator`: Split PDF document into multiple PDF documents • `Clear`: Reset iterator to initial value

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** `de.softproject.integration.adapter.pdf.PDFSplitterIteratorAdapter`: Main class (default)
`Start`	Page number from which to start **Possible values:** Any positive integer
`Step`	Step size by which the `Start` parameter is incremented for the next iteration. **Possible values:** Any positive integer
`Count`	Number of pages to be output as a separate PDF document per iteration. **Possible values:** Any positive integer
`toXML`	Output PDF document Base64-encoded in an XML document **Possible values:** • `yes`: Output XML document • `no`: Output PDF document as binary data stream (default).
`ReadPartially`	Load PDF document into memory in multiple parts (can improve the performance) **Possible values:** • `yes`: Load in several parts one after the other into the working memory • `no`: Load PDF document(s) into memory in one piece (default)

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

Input

This function adapter expects an adapter-specific input XML structure which contains the path to the processed PDF document in the repository or the Base64 string.

PDF in PDF format

The following input XML document contains the path to a PDF document in the repository.

```
<?xml version="1.0" encoding="UTF-8"?>
<PDF url="xstore://Project/Folder/Input.pdf"/>
```

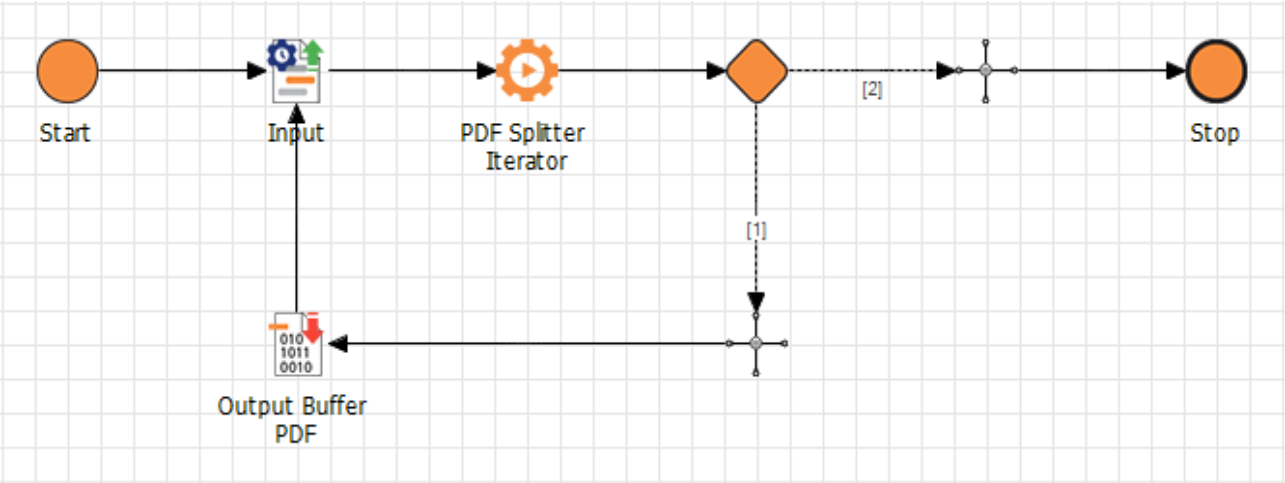
PDF in Base64 format

The following input XML document contains the Base64 string.

```
<?xml version="1.0" encoding="UTF-8"?>
<PDF encoding="base64">
  Base64-Zeichenkette
</PDF>
```

Output

If the `ToXML` parameter is **disabled**, the output can be saved directly as a buffer file in PDF format.

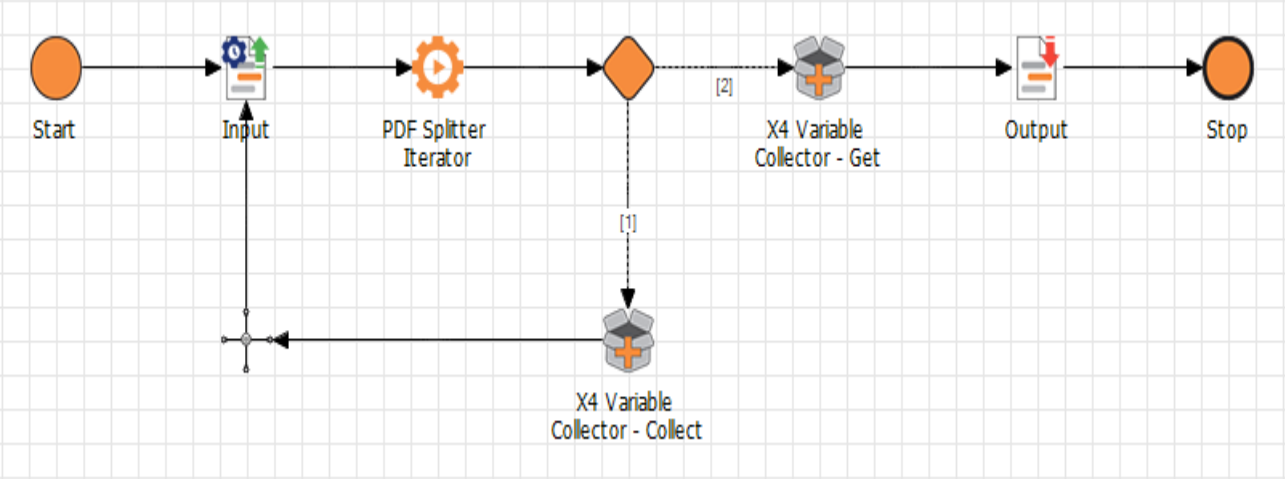


If the `ToXML` parameter is **enabled**, the output looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<PDF encoding="base64">
    Base64-Zeichenkette
</PDF>
```

Example

The process splits the XML and stores all intermediate results in XML format in a variable. To store the variables, the X4 Variable Collector adapter is used.



10.17 Security

Adapter category *Security* contains adapters and connectors that allow to encrypt and decrypt data in Technical Processes and to create and validate signatures.

10.17.1 Antivirus

`Antivirus`: Connects to a ClamAV server, an open source (GPL) antivirus engine. Input files are checked for viruses at adapter runtime.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** *Scan*: Check input files for viruses

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** `de.softproject.integration.adapter.antivirus.AntiVirusAdapter`: Main class (default)

host	IP address or host name of the ClamAV server. **Possible values:** • any valid string, e. g. `localhost`, `127.0.0.1` • *`localhost`*: ClamAV host (default)
port	Port number of the ClamAV server **Possible values:** • any integer number • *`3310`*: FTP port (default)
scanTimeoutInMs	Waiting time for a scan in milliseconds. **Possible values:** • Any integer number. The default value is 500. • A value of 0 does not set a timeout. Please be aware that this may lead to infinite wait times. • Negative values will be ignored and the default value of 500 will be used.

Status values

`1`	The adapter outputs a result document. The operation was successful, no viruses were found *or* the input file is empty.
`-1`	The adapter outputs the input file. The operation could not be executed. Details of the error message can be found in the server logs.
`400`	The adapter outputs a result document. The operation was successful, viruses were found in the input file.

Input

The adapter expects any input file, both binary files and Base64-encoded documents (text or XML files).

Output

If the operation failed, the adapter outputs the input file.

If the operation was successful, the adapter outputs a UTF-8-encoded XML result document of the following structure. If viruses have been found, the value is set to `true`, otherwise the value is set to `false`.

```
<?xml version="1.0" encoding="UTF-8"?>
<ScanResult>
    <!--The value true is output if viruses were found in the input file-->
    <!--The value false is output if no viruses were found in the input file-->
    <VirusFound><!--value--></VirusFound>
</ScanResult>
```

Dependencies and Setup instructions

This adapter uses the following libraries: ClamAV library `clamav-client-1.0.1.jar`

10.17.1.1 Setup ClamAV

For the adapter to work properly, the ClamAV server must be installed and running on the same machine as the X4 BPMS. A detailed installation description and adjustments to the configuration files can be found in the official ClamAV manual (*see chapters 4 and 5*).

> **Download setup packages**
> A list of available installation packages can be found on the provider's website: https://www.clamav.net/downloads#otherversions
>
> - Download version with setup wizard for Windows (32 bit): `clamav-<version_number>-x86.zip`
> - Download version with setup wizard for Windows (64 bit): `clamav-<version-number>-win-x64.zip`

10.17.1.2 Updating the virus database

The settings for automatically updating the virus database are described in the official ClamAV manual (*see chapter 4*).

10.17.2 BCrypt Calculator

`BCrypt Calculator`: Creates and verifies salted BCrypt hashes.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `GenerateHash`: creates a salted hash to a specified plain text password • `Verify`: verifies the plain text password with the entered salted hash • `VerifyAndReplace`: verifies the plain text password with the entered salted hash and creates a new salted hash

Parameter

`Adapter`	Adapter main class (do not change!) **Possible values:** *de.softproject.integration.adapter.bcrypt.BCryptAdapter*: main class (default)
`clearText`	Plain text password **Possible values:** Any string, e. g. *`MyPassword`* ℹ The parameter must not be empty.
`bcryptHash`	Salted BCrypt hash to be compared with the plain text password; Required for the operations `Verify` and `VerifyAndReplace` **Possible values:** Any salted BCrypt hash
`newclearText`	New plain text password; Only required for the operation `VerifyAndReplace` **Possible values:** Any string, e. g. *`MyPassword`* ℹ The parameter must not be empty.

Status values

`1`	The operation was executed successfully and the output contains an XML file with the result
`0`	Operations `Verify` and `VerifyAndReplace`: the parameter `bcryptHash` does not correspond to the `clearText` string
`-1`	An error occurred during the operation's execution • The requited parameters are not specified • Operations `Verify` and `VerifyAndReplace`: the parameter `bcryptHash` does not contain a salted BCrypt hash

Input

The adapter does not expect an input for any operation.

Output

Depending on the operation, the adapter outputs different results:

- Operations `GenerateHash` and `VerifyAndReplace`: XML file with the salted hash
- Operation `Verify`: XML file with the result whether the salted hash matches the plain text password
- Operation `VerifyAndReplace` (if salted hash does not match the plain text password): XML file with the result that the salted hash does not match the plain text password
- XML file with error messages

Sample output

```
<?xml version="1.0" encoding="UTF-8"
<Hash>$2a$10$EcgngKl9hSlppvEcDspLzeq.9Blqim1SFNVzY6tbn5ixTZFMh4xs</Hash>
```

10.17.3 Checksum Creator

? Unknown Attachment `Checksum Creator`: Creates checksums for files in a folder and outputs the checksums together with the file names and file properties as an XML document.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *`Count CRC`*: Use the CRC algorithm to form the checksums. • *`Count Adler`*: Use Adler algorithm to form the checksums

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.checksum.ChecksumAdapter`*: Main class (default)
path	Folder for whose contents checksums are to be created **Possible values:** Any valid character string (e.g. *`%USER_COLLECTION%/Project/Folder`*)
useRecursion	Handle folder recursively **Possible values:** • *`yes`*: Include elements in subfolders • *`no`*: Restrict to the items in the specified folder

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

This adapter does not require any input from the previous process step.

Output

Example of an output XML document with file names, checksums and an included subfolder:

```
<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`: Generates or calculates the hash code (RSA, MD5 etc.) of the input data.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Calculate`*: Calculate hash code

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.digest.DigestAdapter`*: Main class (default)
algorithm	Algorithm used to calculate the hash code **Possible values:** • *`MD5`*: Use *Message Digest 5* algorithm (default) • *`SHA`*: Use *Secure Hash (0)* algorithm • *`SHA-1`*: Use *Secure Hash 1* algorithm • *`SHA-256`*: Use *Secure Hash 256* algorithm • *`CRC`*: Use *Cyclic Redundancy Check* algorithm • *`ADLER`*: Use *Adler* algorithm

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Output

This adapter outputs a *UTF-8*-encoded XML document containing the algorithm used and the calculated hash code:

```
<digest type="Algorithmus"><!--Hashcode der Input-Daten--></digest>
```

For example, the adapter outputs the following XML document:

```
<digest type="MD5">2e5941901e8b5e214dcc5032dfce71bb</digest>
```

10.17.5 JSON WebToken Calculator

`JSON WebToken Calculator`: Generates, verifies and processes JSON WebTokens.

Former name of the adapter until version 5.5.4: *JSON WebToken*

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `CreateJWT`: Generates a JSON WebToken (JWT) from an XML document • `Verify`: Verifies the JWT generated by the X4 Server • `CreateXML`: Created from a JWT generated by the X4 Server the corresponding XML document

Parameters

The following parameters are required for the above operations:

	CreateJWT	Verify	CreateXML
algorithm	required	required	required
useServerSecret		required if `useSecretAsPublicKey` is not used.	
useSecretAsPublicKey		required if `useServerSecret` is not used.	
secret		required if `useSecretAsPublicKey` is used.	
keystoreUrl	required		required

keystoreType	required		required
keystorePassword	required		required
keyAlias	required		required
keyPassword	required		required

`Adapter`	Adapter main class (do not change!) **Possible values:** *de.softproject.integration.adapter.json.webtokenJSONWebTokenAdapter*: Main class (default)
`algorithm`	Algorithm used for the operation **Possible values:** • `HMAC256` • `HMAC384` • `HMAC512` • `RSA256` • `RSA384` • `RSA512` • `ECDSA256` • `ECDSA384` • `ECDSA512`
`useServerSecret`	*Only relevant for HMAC algorithms.* Specifies if an integrated server secret or a custom server secret is used. **Possible values:** • `true`: Use integrated server secret • `false`: Use custom server secret ⚠ If a custom server secret is used, it must be defined by parameter `secret`.
`useSecretAsPublicKey`	*Only relevant for RSA and ECDSA algorithms.* Determines if the value of the parameter `secret` is used as public key. **Possible values:** • `true`: Use value of parameter `secret` as public key ⚠ If the value of the parameter `secret` is used as public key, then the parameter `secret` is required. • `false`: Don't use value of parameter `secret` as public key

`secret`	*Only relevant for HMAC algorithms.* Password used to encrypt or decrypt a token. **Possible values:** Base64 encoded password
`keystoreUrl`	URL to the keystore that contains the RSA/ECDSA key pair **Possible values:** URL to a keystore ⚠ The keystore must be supported by the Java Security API, e.g. JKS or PKCS#12.
`keystoreType`	Type of the keystore that contains the RSA/ECDSA key pair **Possible values:** Type of the keystore, e.g. `JKS` or `PKCS12`
`keystorePassword`	Password of the keystore that contains the RSA/ECDSA key pair **Possible values:** Password
`keyAlias`	Alias of the asymmetric key used for RSA/ECDSA algorithms **Possible values:** Alias of the used key
`keyPassword`	Password of the asymmetric key used for RSA/ECDSA algorithms **Possible values:** Password of the used key

Status values

`1`	The adapter operation was executed successfully: • Operation `CreateJWT`: A JSON WebToken is stored in the output • Operation `Verify`: The JSN WebToken is valid • Operation `CreateXML`: The XML document for the JSON WebToken is stored in the output
`-1`	An error occurred during the execution of the adapter: • Operation `CreateJWT` and `CreateXML`: An error message is stored in the output • Operation `Verify`: The JSON WebToken is invalid or expired
`400`	The token is expired. ⓘ This status value is only output by the operations `CreateXML` and `Verify`.

Input

The adapter expects different inputs depending on the operation:

- Operation `CreateJWT`: An XML document with the following format is expected:

- The Header types `typ` (parameter: `keystoretype`) and `alg` (parameter: `algorithm`) is set by the adapter.
- Do not use claims with duplicated names.

The Header element is optional and can be declared when needed. For example, Microsoft Azure expects a header declaration.

Sample 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>
```

For additional information on the claims to be used and their XML objects, see iana.org. For non-reserved claims, the names can be arbitrary, but they must not be the same as already reserved claims such as *`bsp.`* or *`exp.`*

The values *`String`*, *`Boolean`*, *`Integer`*, *`Long`* and *`Double`* are available as `type`.

The elements `ExpiresAt`, `NotBefore` and `IssuedAt` expect information in Unix time format.

- Operations `Verify` and `CreateXML`: A JSON WebToken is expected, which can be taken from an intermediate step or passed as a text file (`.txt`).

Output

The adapter outputs different results depending on the operation:

- Operation `CreateJWT`: A JSON WebToken is stored in the output
- Operation `Verify`: If the JWT is valid, it is output
- Operation `CreateXML`: The XML document for the JSON WebToken is stored in the output

10.17.6 LDAP Connector

`LDAP Connector`: Establishes a connection to an LDAP server (*Lightweight Directory Access Protocol*) to query contact data - if required via LDAP Secure. Various operations can be executed via adapter-specific XML documents as input data.

> The configuration for LDAP Secure with self-signed certificates can be done in the configuration file `X4config.xml`, see LDAPS configuration.

Properties

`Operation`	Determines which operation the adapter executes **Possible values:** *`Execute`*: Execute adapter

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.ldap.Ldap`*: Main class (default)
`java.naming.factory.initial`	Name of the class providing the LDAP service **Possible values:** Any class name, e.g. *`com.sun.jndi.ldap.LdapCtxFactory`*
`java.naming.provider.url`	URL of the LDAP(S) server **Possible values:** Any URL
`java.naming.security.authentication`	Authentication method **Possible values:** Any valid string (e.g. *`simple`*)
`java.naming.security.principal`	Name of the user or authentication application (depending on the authentication method). **Possible values:** Any valid string (e.g. *`cn=S. User, ou=NewHires, o=JNDITutorial`*)
`java.naming.security.credentials`	Associated password of the user or authentication application (depending on the authentication type). **Possible values:** Any string (e.g. *`mysecret`*)

`java.naming.referral`	Defines how redirects should be handled. **Possible values:** • `ignore`: Ignore redirects • `follow`: Automatically follow all redirects • `throw`: Throw a ReferralException for each forwarding (see Throwing a ReferralException)
`attributesAsElements`	Output attributes of an object **Possible values:** • `true`: Output attributes of an object • `false`: Do not output object attributes (default)
`acceptAllCertificates`	Disables the security settings by accepting any certificate. **Possible values:** • `true`: Any certificate will be accepted. • `false`: Only self-signed certificates will be accepted (default).
`acceptAllHostnames`	Disables the security settings by accepting any host name contained in the certificate. **Possible values:** • `true`: Accepts any host name. • `false`: Security settings are disabled (default).

Status values

`1`	The operation was executed successfully.
`2`	*Operation `Compare` required*. LDAP entry matches the information in the input XML document.
`3`	*Operation `Compare` required*. LDAP entry does **not** match the information in the input XML document.
`-1`	The operation failed due to a technical error.

Throw ReferralException

If redirects are to be processed manually, then the following steps must be followed:

1. Catch exception.
2. Read forwarding information by using `ReferralException.getReferralInfo()` and e.g. ask the user if the forwarding should be followed.
    a. If the referral is to be followed, read the referral context with `ReferralException.getReferralContext()` and call the original context method again with the same arguments that should be used for the original call.
    b. If the referral is not to be followed, call `ReferralException.skipReferral()`.

i. If this method returns `true`, then it means that multiple redirects should be followed.
   If that is the case, call `ReferralException.getReferralContext()` to continue.
   If a context method calls the result, then an exception is thrown again for the next redirect. Handle this exception from step 1.
ii. If the method returns `false`, then there are no more references and this procedure can be terminated.

More information at https://docs.oracle.com/javase/jndi/tutorial/ldap/referral/throw.html.

Input

This function adapter expects specific input XML structures that contain LDAP commands. These input XML structures can be generated dynamically via XSL mappings depending on the processed data.

10.17.6.1 search command

You can use the LDAP `search` command to search for entries in the LDAP directory.

Input

The *LDAP Adapter* expects the following input XML structure to execute the `search` command:

```
<?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>
```

Output

The successfully executed adapter outputs the following XML document:

```
<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 command

You can use the LDAP add command to add entries to the LDAP directory.

Input

The *LDAP Adapter* expects the following input XML structure to execute the add command:

```
<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>
```

Output

The successfully executed function adapter does not output an XML document after the add command is executed, but only the status `1 (successful)` or in case of error the status `-1 (Error)`.

10.17.6.3 modify command

With the LDAP command `modify` you can change entries in the LDAP directory. Thereby, as in a transaction, either all entries are changed or, in case of an error, no changes are made to the entries.

Input

The *LDAP Adapter* expects the following input XML structure to execute the `modify` command:

```
<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>
```

Example of a password `replace`:

```
<ldap>
    <modify>
        <name>dc=sp-linux</name>
        <add>
            <o>Tester</o>
        </add>
        <replace>
            <unicodePW base64="true">dGVzdCE=</unicodePW>
        </replace>
    </modify>
</ldap>
```

Output

The successfully executed function adapter does not output an XML document after the `modify` command is executed, but only the status `1 (successful)` or in case of error the status `-1 (Error)`.

10.17.6.4 modifyDN (RDN) command

With the LDAP command `modifyDN (RDN)` you can edit entries in the LDAP directory.

Input

The *LDAP Adapter* expects the following input XML structure to execute the `modifyDN RDN` command:

```
<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>
```

Output

The successfully executed function adapter does not output an XML document after the `modifyDN RDN` command is executed, but only the status `1 (successful)` or in case of error the status `-1 (Error)`.

10.17.6.5 delete command

You can use the LDAP `delete` command to delete entries in the LDAP directory.

Input

The *LDAP Adapter* expects the following input XML structure to execute the `delete` command:

```
<ldap>
    <delete>
        <name>sn=Tester, o=SoftProject, dc=sp-linux</name>
        <!--dn-->
    </delete>
</ldap>
```

Output

The successfully executed function adapter does not output an XML document after executing the `delete` command, but only status `1 (successful)` or in case of error status `-1 (Error)`.

10.17.6.6 compare command

You can use the LDAP `command` to compare entries in the LDAP directory.

Input

The *LDAP Adapter* expects the following input XML structure to execute the `compare` command:

```
<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>
```

Output

The successfully executed function adapter does not output an XML document after executing the `compare` command, but only the `CompareTRUE` status if the LDAP entry matches the information in the input XML document, or the `CompareFALSE` status otherwise.

10.17.7 Password Generator

`Password Generator`: Generates a random password on the basis of a password policy defined by the adapter's parameter. This adapter does not expect an input and outputs an XML document containing the generated password.

Properties

Operation	Defines this adapter's operation **Possible values:** `Generate`: Generate a password

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.adapter.security.PasswordGenerator`*: Main class (default)

length	Number of characters for the generated password **Possible values:** • Any positive integer number • `8`: Password with 8 characters (default)
numberRequired	Defines if the password will contain a numerical character **Possible values:** • *`True`*: Password will contain a number • *`False`*: Password will not contain a number (default)
lowerCaseCharRequired	Defines if the password will contain a lowercase character **Possible values:** • *`True`*: Password will contain a lowercase character • *`False`*: Password will not contain a lowercase character (default)
upperCaseCharRequired	Defines if the password will contain an uppercase character **Possible values:** • *`True`*: Password will contain an uppercase character • *`False`*: Password will not contain an uppercase character (default)
specialCharRequired	Defines if the password will contain a special character (*@#$%*) **Possible values:** • *`True`*: Password will contain a special character • *`False`*: Password will not contain a special character (default)

Status values

`1 (successful)`	The operation has been executed successfully.
`-1 (failed)`	The operation could not be executed successfully due to a technical error.

Input

This adapter does not expect an input.

Output

This adapter outputs an XML document with the following structure:

```
<Password><!-- Generated password --></Password>
```

10.17.8 Random Number Generator

`Random Number Generator`: Creates random numbers based on the specified configuration.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Generate`: generates an XML document with one or more random numbers • `SetRandomNumbers`: generates random numbers into an existing XML document via XPath

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.randomNumberGenerator.RandomNumberGenerator*: Main class (default)
`min`	Minimum value from which a random number is to be generated **Possible values:** any positive number, e.g. *`1`* ℹ Mustn't be negative or *`0`*.
`max`	Maximum value from which a random number is to be generated **Possible values:** any positive number, e.g. *`100`* ℹ Mustn't be negative or *`0`*.
`count`	Number of random numbers to be generated **Possible values:** any positive number, e.g. *`4`* ℹ Mustn't be negative or *`0`*.
`XPath`	XPath to the desired XML element ℹ Only important for the operation `setRandomNumbers`. **Possible values:** any XPath expression, e.g. *`Data/Example`*

`elementName`	Name of the XML element in which the random number is to be generated. Only important for the operation `setRandomNumbers`. **Possible values:** Any name for an XML element, e.g. *`Id`*.

Status values

`1`	The adapter operation was executed successfully and the output contains an XML document with the random number(s)
`-1`	An error occurred during execution • Operations `Generate` and `setRandomNumbers`: the parameter `count` is less than `1` • Operation `setRandomNumbers`: the input is no XML document

Input

The adapter expects an XML document as input for the operation `SetRandomNumbers`.

Output

Depending on the operation, the adapter outputs the following results:

- Operation `Generate`: XML document with the generated random number(s)
- Operation `SetRandomNumbers`: XML document with added random number(s)

Example

```
<?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`: Encrypts or decrypts input data (byte data stream) with the RSA algorithm and outputs a byte data stream (2048-bit RSA key).

Former designation until version 4.5.1: *RSA Encryption Adapter*

The maximum byte length of encrypted input data is the length of the RSA key in bits divided by 8, minus 11 fill bytes. For example, the maximum byte length for a 1024-bit RSA key is calculated as follows: 1024/8 -11 = 117 bytes; A 2048-bit RSA key decrypts input data with a maximum length of 245 bytes.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *`Encrypt`*: RSA-encrypt input data • *`Decrypt`*: Decrypt RSA-encrypted input data

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *en.softproject.integration.adapter.rsa.RSAAdapter*: Main class (default)
path	Path to the keystore file **Possible values:** Any absolute or relative path
alias	Short name of the entry that has a key in the keystore **Possible values:** Any string
keyType	Key type used **Possible values:** • *private*: Use private key (default) • *public*: Use public key
keystorePassword	Password for the keystore file **Possible values:** Any string
privatePassword	Password to obtain the private key from the keystore file **Possible values:** Any string

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.17.10 SAML Calculator

`SAML Calculator`: Signs, validates, encrypts and decrypts SAML assertions and requests.

Properties

Operation	Defines the operation executed by the function adapter. **Possible values:** • `Encrypt Assertion`: Encrypt SAML assertion **Required parameters:** `cryptAlias`, `cryptPassword` • `Decrypt Assertion`: Decrypt SAML assertion **Required parameters:** `cryptAlias`, `cryptPassword` • `Sign Assertion`: Sign SAML assertion with a certificate **Required parameters:** `signatureAlias`, `signaturePassword`, `algorithm` • `Validate Assertion`: Validate SAML assertion **Required parameters:** `cryptAlias`, `cryptPassword`, `signatureRequired`, `skipValidation` • `Sign Request`: Sign SAML request with a certificate **Required parameters:** `signatureAlias`, `signaturePassword`, `algorithm` • `Validate Request`: Validate SAML request **Required parameters:** `signatureRequired`, `skipValidations`

Parameters

`Adapter`	Main adapter class (Do not change!) **Possible values:** *de.softproject.x4.adapter.saml.SamlAdapter*: Main class (default)
`keystoreUrl`	URL of the keystore that is used to encrypt and decrypt SAML assertions. **Possible values:** String (URL)
`keystorePassword`	Password of the keystore that is used to encrypt and decrypt SAML assertions. **Possible values:** String
`keystoreType`	Type of the keystore that is used to encrypt and decrypt SAML assertions. **Possible values:** Type of the keystores, e.g. `JKS`, `PKCS12`, ...
`signatureRequired`	Defines if a signature is required to decrypt a file. **Possible values:** `true` / `false`

`skipValidations`	Defines if the signature and validity verifications are skipped during decrypting. **Possible values:** `true` / `false`
`cryptAlias`	Alias of the key pair or certificate that is used to decrypt and encrypt SAML assertions.
`cryptPassword`	Passwort of the key pair that is used to decrypt SAML assertions.
`signatureAlias`	Alias of the key pair that is used to sign SAML assertions.
`signaturePassword`	Passwort of the key pair that is used to sign SAML assertions.
`algorithm`	Algorithm that is to be used to sign SAML assertions. **Possible values:** • `SHA1`: • `SHA256`: • `SHA512`:

Status values

1	The adapter outputs a result.
-1	An error occurred during the adapter's execution (for details see server log).

Input

The adapter expects a SAML assertion or a SAML request depending on the operation to be executed.

- SAML Assertion:

Example 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="_be9967abd904ddca
e3c0eb4189adbe3f71e327cf93">
        <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:

Example 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>
```

Output

The Adapter outputs different files depending on the operation executed:

- `Encrypt Assertion`: The adapter outputs a decrypted SAML assertion.
- `Sign Assertion`: The adapter outputs a signed SAML assertion.
- `Sign Request`: The adapter outputs a signed SAML request.

10.17.11 SSH Connector

`SSH Connector`: Connects to SSH servers using the secure shell protocol

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Execute`: Executes the input command on the server

Parameters

`Adapter`	Adapter main class (do not change!) **Possible values:** *de.softproject.integration.adapter.ssh.SSHAdapter*: main class (default)
`host`	Server host **Possible values:** Any valid host or IP, e. g. *`localhost`* or any other IP direction

`user`	User to log in to the server **Possible values:** String containing the user name, e. g. *`userName`*
`password`	Password to log in to the server **Possible values:** String containing the password, e. g. *`myPassword`*
`port`	Server port **Possible values:** • `22` (default) • any other Integer with the port
`timeout`	Timeout in milliseconds for waiting until the connection to the server is established **Possible values:** • 30000 ms (default) • any other Integer with the time in milliseconds

Status values

`0`	The operation was executed successfully.
Distinct to `0`	An error occurred during the operation's execution

Input

The adapter expects a specific XML document as input:

Sample input

```
<?xml version="1.0" encoding="UTF-8"
<Root>
    <Command>netstat -plnt |grep 8080</Command>
</Root>
```

ⓘ The element `<Command>` contains the command to be executed on the server.

Output

The adapter returns an XML document with the output and the exit status.

Output corresponding to the sample 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`: Generates a TOTP one-time password compatible with Google Authenticator and others tools.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `GenerateSecret`: Create a 20-byte BASE32 character string which can be used as a secret key for password generation or validation, or as base for a QR code image • `GeneratePassword`: Create a one-time password based on the secret key provided in the input XML document • `VerifyPassword`: Verify a one-time password in combination with the secret key – both provided in the input XML document • `GenerateAuthenticatorQRCode`: Create a `512x512 px` image (`.png`) based on the secret key provided in the input XML document ⓘ This operation also uses the parameters `issuer` and `account` to annotate the QR code. This information will be visible when the code is scanned by Google Authenticator or a similar tool.

Parameters

`Adapter`	Main class of the adapter (Do not change!) **Possible values:** *de.softproject.x4.adapter.totpgenerator.TOTPGenerator*: Main class (default)
`issuer`	Company or organization name issuing the one-time passwords ⓘ This parameter should be treated like a display name, since it will be shown in the authenticator app as provided, e.g. `SoftProject GmbH`.

`account`	Account which is tied to the secret key – typically a username or e-mail address ⓘ This parameter should be treated like a display name, since it will be shown in the authenticator app as provided, e.g. `John Doe` or `john.doe@example.org`.

Status values

`1`	The operation was executed successfully.
`-1`	An error occurred during the operation's execution.
`403`	The password does not match, e.g. because the input structure is not well-formed

Input

The adapter expects different input structures depending on the selected operation.

- **Operation `GenerateSecret`:**
  This operation doesn't require any 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>
```

Output

The adapter outputs different structures depending on the selected operation.

- **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`:**
  An image (`.png`) with the QR code is output.

10.17.13 UUID Generator

`UUID Generator`: Generates universal unique identifiers (UUIDs).

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Generate`: generates an XML document with one or several UUIDs • `SetUUID`: generates UUIDs in an already existing XML document using XPath UUIDs

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.uuid.UuidGeneratorAdapter*: Main class (default)
`count`	Number of the UUIDs to be generated Only important for the operation `Generate` **Possible values:** Any positive number, e. g. `1` Mustn't be negative or 0.
`generateBraces`	Set generated UUIDs in braces **Possible values:** • `true`: sets generated UUID in braces • `false`: doesn't generate braces
`xPath`	XPath to the desired XML element Only important for the operation `SetUUID` **Possible values:** Any XPath expression, e. g. `Data/Beispiel`
`elementName`	Name of the XML element containing the UUID to be generated. The element may already exist and will be overwritten, if necessary. Only important for the operation `SetUUID` **Possible values:** Any XML element name, e. g. `Id`

Status values

`1`	The adapter has been executed successfully and the output contains an XML document with the UUID(s)
`-1`	The adapter execution failed **Possible reasons:** • Operation `Generate`: the parameter `count` is smaller or equal `0` • Operation `SetUUID`: the input is not an XML document

Input

The adapter expects an XML document for the operation `Generate`.

Output

The output depends on the selected operation:

- `Generate`: the adapter outputs an XML document with the generated UUID(s)
- `SetUUID`: the adapter outputs an XML document with the added UUID(s)

Sample output

```
<?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):` Encrypts or decrypts any XML document according to the W3C standard XML-Encryption (XML-Enc) based on a *PKCS12* or *JCEKS* keystore. This adapter supports the AES encryption algorithm with 128, 192 or 256 Bit key length as well as the Triple-DES algorithm. For encryption/decryption based on a password, see XML Encryption (Password).

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *`Encrypt`*: Encrypt input XML document • *`Decrypt`*: Decrypt input XML document
Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.xml.encryption.XmlEncryption*: Main class (Default)

method	Algorithm used for encryption/decryption **Possible values:** • `AES_128`: Use AES algorithm with 128 Bit key length • `AES_192`: Use AES algorithm with 192 Bit key length • `AES_256`: Use AES algorithm with 256 Bit key length • `TRIPLEDES`: Use Triple-DES algorithm
xpath	Path to the element(s) to be encrypted/decrypted addressed using an XPath expression **Possible values:** Any valid XPath expression, e.g. `//Data`
scope	Defines the encryption scope within the XML document structure that matches the XPath expression • `Element`: Encrypt the whole element • `Content`: Encrypt only the element content
keyStoreURL	URL to a valid keystore document **Possible values:** Any valid URL, e.g. `x4db:/1/Project/MyKeystore.pkcs12`
keyStoreType	Keystore type to be used **Possible values:** • `jceks`: Use keystore in the JCEKS format • `pkcs12`: Use keystore in the PKCS12 format
password	Password to access the keystore **Possible values:** Any string
alias	Alias for the key inside the keystore **Possible values:** Any string
aliasPassword	Password to retrieve the key via its alias specified above (optional) **Possible values:** • Any string • `(empty)`: Use keystore password (default)

Status values

-1 (error)	An error occurred during the adapter's execution (see *X4 Server* log). In case of an error, this adapters also outputs a text file that contains the corresponding exception.
0 (empty)	Document has not been changed. No matching elements were found.
1 (successful)	Document has been processed, and all matching elements have been encrypted/ decrypted.

Example

In the following XML document, all `Data` elements are to be encrypted using XML-Encryption with the AES128 algorithm:

Input

```
<RootElement>
  <Data>Hello</Data>
  <Data>World!</Data>
  <Public>© SoftProject GmbH</Public>
</RootElement>
```

As an example, the adapter returns the following encrypted result XML document:

Result

```
<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)`: Encrypts or decrypts any XML document according to the W3C standard XML-Encryption (XML-Enc) based on a password. This adapter supports the AES encryption algorithm with 128, 192 or 256 Bit key length as well as the Triple-DES algorithm. For encryption/decryption based on a keystore, see XML Encryption (Keystore).

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *`Encrypt`*: Encrypt input XML document • *`Decrypt`*: Decrypt input XML document

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.xml.encryption.XmlEncryption*: Main class (Default)
method	Algorithm used for encryption/decryption **Possible values:** • `AES_128`: Use AES algorithm with 128 Bit key length • `AES_192`: Use AES algorithm with 192 Bit key length • `AES_256`: Use AES algorithm with 256 Bit key length • `TRIPLEDES`: Use Triple-DES algorithm
xpath	Path to the element(s) to be encrypted/decrypted addressed using an XPath expression **Possible values:** Any valid XPath expression, e.g. `//Data`
scope	Defines the encryption scope within the XML document structure that matches the XPath expression • `Element`: Encrypt the whole element • `Content`: Encrypt only the element content
password	Password for encryption/decryption **Possible values:** Any string

Status values

-1 (error)	An error occurred during the adapter's execution (see *X4 Server* log). In case of an error, this adapters also outputs a text file that contains the corresponding exception.
0 (empty)	Document has not been changed. No matching elements were found.
1 (successful)	Document has been processed, and all matching elements have been encrypted/ decrypted.

Example

In the following XML document, all `Data` elements are to be encrypted using XML-Encryption with the AES128 algorithm:

Input

```
<RootElement>
  <Data>Hello</Data>
  <Data>World!</Data>
  <Public>© SoftProject GmbH</Public>
</RootElement>
```

As an example, the adapter returns the following encrypted result XML document:

Result

```
<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

? Unknown Attachment `XML Signature`: Adds a signature to an XML document or verifies a signature within an XML document based on a keystore file in PKCS#12 format.
Former designation until version *4.6: Digital Signature Adapter*

Properties

Custom Parameters	Set user-defined parameters; see Using Component Parameters

Operation	Determines which operation the adapter executes **Possible values:** • `Sign`: Sign XML document • `Check signature`: Check the validity of the signature

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.digitalsignature.DigitalSignatureAdapter`*
keyStore	Path to a keystore file in PKCS#12 format **Possible values:** • Relative or absolute path to a PKCS#12 keystore • URL to a PKCS#12 keystore • *Example: `%USER_COLLECTION%/Project/Folder/Keystore.p12` or `xstore://XMLSignatureXstore/Resources/KeyStore.p12`*
ksPass	Keystore password **Possible values:** Any string
keyPass	Password for a key (optional) **Possible values:** Any string

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

The adapter expects any XML document as input for the `Sign` operation.

Output

The adapter outputs a signed XML document for the `Sign` operation.

10.17.16 AES Encryption

`AES Encryption`: Encrypts and decrypts input data using the AES encryption algorithm.

AES Encryption
The publicly available encryption algorithm **Advanced Encryption Standard (AES)** is one of the most popular and secure encryption methods.
It is based on a series of byte substitutions, permutations and linear transformations that are performed on 16-byte data block – therefore the term block encryption. These operations are repeated several times, and in each of these iterations an individual iteration key calculated from the actual key is used in the calculations. If only a single bit in the key or in the data block is changed, a completely different cipher block is created – an advantage over classical encryption methods.
The designations `AES-128`, `AES-192` and `AES-256` specify the key length: 128, 192 and 256 bits. Until now, no practical attack is known for any of the AES variants. AES is therefore the preferred encryption standard.

You can find additional information on AES encryption under https://www.comparitech.com/blog/information-security/what-is-aes-encryption/.

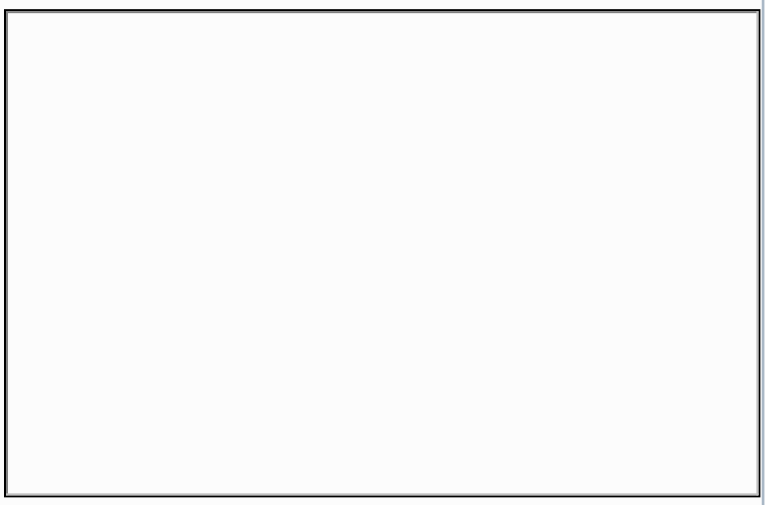
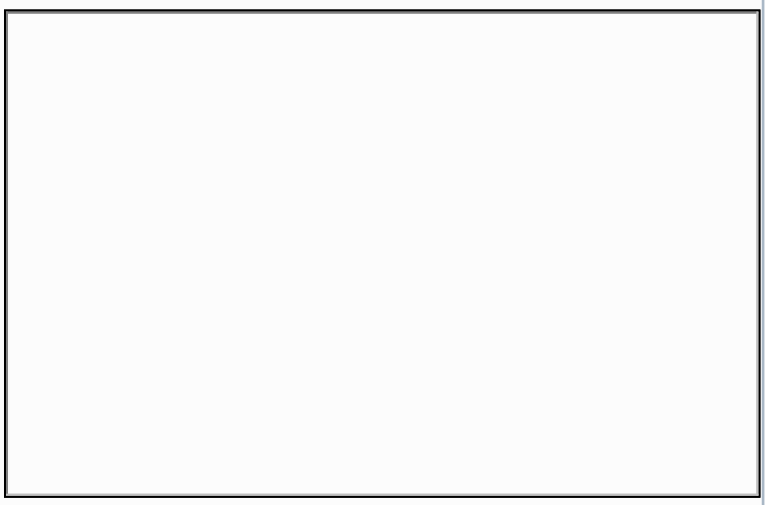
Properties

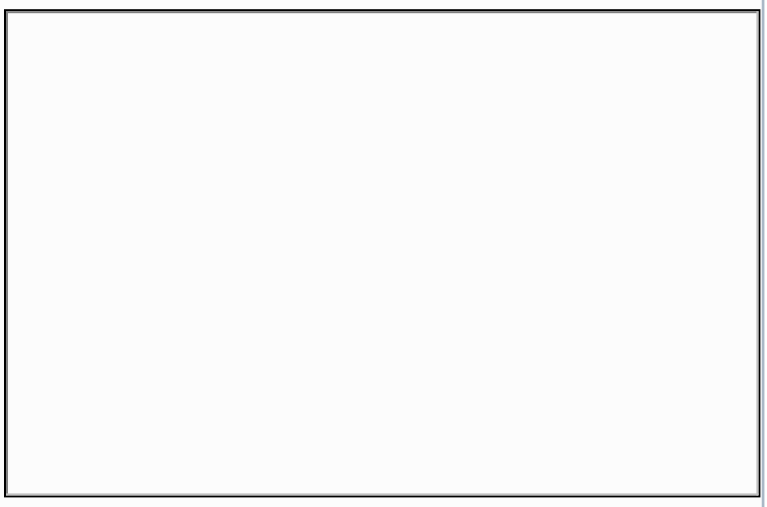
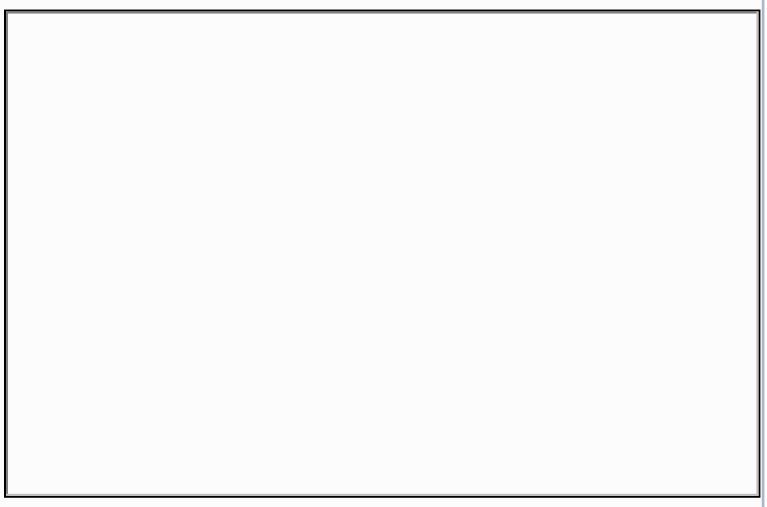
`Operation`	Defines the operation executed by the adapter **Possible values:** • `Encrypt`: Encrypts input data with the AES encryption algorithm • `Decrypt`: Decrypts input data with the AES encryption algorithm • `GenerateSharedKey`: Generates a shared key to encrypt and decrypt the input data • `GenerateInitializationVector`: Generates a random initialization vector to encrypt and decrypt the input data

Parameters

`Adapter`	Adapter main class (do not change!) **Possible values:** *`de.softproject.integration.adapter.aes.AesEncryption`*: main class (default)
`password`	*Required for operation `GenerateSharedKey`*. Password to create the shared key.

`salt`	*Required for operation `GenerateSharedKey`*. Salt to create the shared key (Base64 encoded). In cryptography, salt refers to a randomly selected character string that is appended to a given plaintext before further processing (e.g. input to a hash function) to increase the entropy of the input. It is often used for storing and transmitting passwords in order to increase information security.
`keyAlgorithm`	*Required for operation `GenerateSharedKey`*. Algorithm to create the shared key. **Possible values:** • `DESede`: Generates secret keys for use with the DESede (Triple DES) algorithm. • `PBEWith<digest>And<encryption>` respectively `PBEWith<prf>And<encryption>`: Secret key factory for use with PKCS5 password-based encryption, where `<digest>` represents a message digest, `<prf>` a pseudo-random function and `<encryption>` an encryption algorithm, e.g. *`PBEWithMD5AndDES`* (PKCS #5, version 1.5), *`PBEWithHmacSHA256AndAES_128`* (see PKCS #5, Version 2.0). • `PBKDF2With<prf>`: Password-based key derivation algorithm (see PKCS #5, Version 2.0) using the pseudo-random function `(<prf>)`, e.g. *`PBKDF2WithHmacSHA256`*
`keyLength`	*Required for operation `GenerateSharedKey`*. Key length to generate the shared key. **Possible values:** • `128` • `196` • `256`
`keyIteration`	*Required for operation `GenerateSharedKey`*. Iterations to be used in order to create the shared key.

`cipherAlgorihm`	Algorithm to encrypt and decrypt the input data. **Possible values:** • `AES/CBC/PKCS5PADDING` • `AES/CBC/NOPADDING` • `AES/ECB/PKCS5PADDING` • `AES/ECB/NOPADDING` **Additional information: CBC Mode and ECB Mode** • **Cipher Block Chaining Mode (CBC Mode)** is a mode in which block ciphers can be operated. Before encrypting a plaintext block, it is first linked to the ciphertext block created in the previous step by XOR (exclusive OR). The structure of encryption and decryption in CBC mode is shown in the following figures:  

	• **Electronic Code Book Mode (ECB Mode)** is the simplest operating mode for block ciphers, since each plaintext block is encrypted separately. Thus, using the same key, the same plaintext blocks always result in the same ciphertext block. This is also the big disadvantage of this method, because plaintext patterns are preserved. The following figures show the structure of the ECB for encryption and decryption:  
`sharedKey`	Shared key to encrypt and decrypt input data (Base64 encoded).

`initializationVector`	Initialization vector to encrypt and decrypt the input data; Works only in CBC mode. ⓘ When encrypting messages, it is important to avoid that the same plaintext blocks always result in the same ciphertext blocks. *Example*: A formal letter usually begins with "Dear Mrs./Mr." followed by the name. Knowing this, an attacker could try to draw conclusions about the key used. To avoid this, the first plaintext block is linked to an initialization vector by XOR. Since the initialization vector was generated randomly, the resulting ciphertexts differ even if the plaintexts begin with identical data.
`prependInitializationVector`	Determines whether the initialization vector should be used as prefix for the encrypted data. ⓘ The initialization vector can be transferred to the encrypted data in plaintext without any problems, since attackers cannot obtain any information from it without the key. The initialization vector is only intended to dilute the beginning of the message before the message is encrypted. If the initialization vector is not transmitted, it must be obtained by other means, since it is necessary for decryption. ▪ **Possible values:** `true` / `false` specification
`encoding`	*Required for operation* `Decrypt`. Specifies the encoding of the decrypted data.
`mimeType`	*Required for operation* `Decrypt`. Specifies the Mime type of the decrypted data.

Status values

`-1`	An error occurred during the operation's execution. For more details consult the server log.
`1`	The operation was executed successfully.

Input

The adapter expects different input documents depending on the selected operation.

- For operation`Encrypt` the adapter expects any input document, e.g. XML, text or binary files.

Sample input for operation Encrypt

```
This is a top secret message and should be AES encrypted!
```

- For operation `Decrypt` the adapter expects AES-encrypted data (Base64).

Sample input for operation Decrypt

```
dpi4dqEK7yiRcyz1HdLs9yxqNLIg6u8fqGR0JxADiCdHb/
6PGDqSxzID6VcuecxUP44x7bkHTggDpn8DO/bgKg==
```

Output

The adapter outputs different result documents depending on the selected operation.

- For operation `Encrypt` the adapter returns the encrypted input data (Base64 encoded).

Sample output for operation Encrypt

```
dpi4dqEK7yiRcyz1HdLs9yxqNLIg6u8fqGR0JxADiCdHb/
6PGDqSxzID6VcuecxUP44x7bkHTggDpn8DO/bgKg==
```

- For operation `Decrypt` the adapter decrypts the encrypted input data.

Sample output for operation Decrypt

```
This is a top secret message and should be AES encrypted!
```

- For operation `GenerateSharedKey` the adapter returns the shared key with the key parameters defined in the adapter.

Sample output for operation GenerateSharedKey

```
SfQYr8UlXmBIeHbZE28XkquKZHFX0qfsSTGBl+4WQDM=
```

- For operation`GenerateInitializationVector` the adapter returns a random initialization vector depending on the cipher algorithm.

Sample output for operation GenerateInitializationVector

```
JXAxGJ343MMjwLFi78z8xw==
```

Sample Scenario:

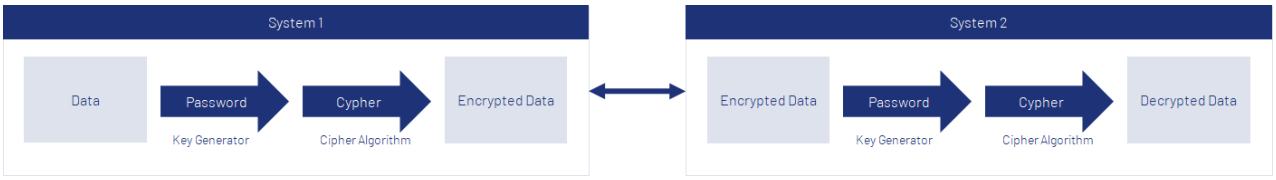
A typical scenario is the exchange of encrypted data between two systems. One way to achieve this is to use a secret password to encrypt and decrypt the data, which is known to both systems. A key is generated from the password using the `GenerateSharedKey` algorithm. The key prepares the password cryptographically and ensures a stronger encryption depending on the selected algorithm. The stronger the key, the more difficult for third parties to decrypt the encrypted data.

As an alternative to the password, also the key can be provided to both systems. This approach is less secure since the additional information about the password with which the key was generated is not known.

Using the key and the cipher algorithm the data is made unrecognizable for third parties and can only be made recognizable again using the same key.

The encrypted data can now be transferred to the second system, either via a network or hardware (USB stick). Thus, no third party can use the data.

The second system also generates the key with exactly the same parameters as the first system. Together with the same cipher algorithm, the data is decrypted and can be sed.



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`: Executes a JMeter test plan and returns a test result as an XML document.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Execute`*: Execute JMeter test plan

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.jmeter.JMeterAdapter`*: Main class (default)
jmeterHome	Absolute path to the JMeter installation folder **Possible values:** Any path (e.g. *`c:\jakarta-jmeter-2.3.1\`*)
proxyHost	IP address or host name of a proxy server for JMeter **Possible values:** Any valid string (e.g. *`192.168.1.1`*)
proxyPort	Port number of the proxy server for JMeter **Possible values:** Any integer
proxyUser	User name of the proxy server for JMeter **Possible values:** Any string
proxyPass	Password of the proxy server for JMeter **Possible values:** Any string
outputFormat	Output format for the test result document **Possible values:** • *`XML`*: Output XML document (standard) • *`CSV`*: Output "Comma separated Value" text file
timestamp	Output timestamp information as attribute ts in the test result document **Possible values:** • *`ms`*: Output timestamp information in milliseconds (default) • *`none`*: Do not output timestamp

dataType	Output data type as attribute dt in the test result document **Possible values:** • *yes*: Output data type • *no*: Do not output a data type
label	Output labelling (label) as attribute lb in the test result document **Possible values:** • *yes*: Output caption • *no*: Do not output a label (default)
responseCode	Output response code as attribute rc in the test result document **Possible values:** • *yes*: Output response code (default) • *no*: Do not output a response code
responseData	Output response data as attribute responseData in the test result document **Possible values:** • *yes*: Output response data • *no*: Do not output response data (default)
responseDataOnError	Output response data in case of errors as attribute responseDataOnError in the test result document **Possible values:** • *yes*: Output response data in case of errors • *no*: Do not output response data in case of errors (default)

responseMessage	Output response message as attribute rm in the test result document **Possible values:** • *yes*: Output response message • *no*: Do not issue a response message (default)
successful	Output success message as attribute successful in the test result document **Possible values:** • *yes*: Output success message (default) • *no*: Do not issue a success message
time	Output time (in milliseconds) as attribute t in the test result document **Possible values:** • *yes*: Output time (default) • *no*: Do not output a time
subresults	Output subresult as attribute subresults in the test result document **Possible values:** • *yes*: Output sub-result • *no*: Do not output a sub-result (default)
assertions	Output statements as attribute assertions in the test result document **Possible values:** • *yes*: Issue statements • *no*: Do not issue statements (default)
latency	Output latency as attribute l in the test result document **Possible values:** • *yes*: Output latency • *no*: Do not output latency (default)

samplerData	Output sampler data as attribute samplerData in the test result document **Possible values:** • *yes*: Output sampler data • *no*: Do not output sampler data (default)
responseHeaders	Output response header as responseHeaders attribute in test result document **Possible values:** • *yes*: Output response header • *no*: Do not output response headers (default)
requestHeaders	Output request header as attribute requestHeaders in test result document **Possible values:** • *yes*: Output request header • *no*: Do not output request headers (default)
encoding	Output character encoding as attribute c in the test result document **Possible values:** • *yes*: Output character encoding • *no*: Do not output character encoding (default)
bytes	Output bytes as attribute bytes in the test result document **Possible values:** • *yes*: Output bytes • *no*: Do not output bytes (default)
url	Output URL as attribute url in the test result document **Possible values:** • *yes*: Output URL • *no*: Do not output URL (default)

threadCounts	Output number of threads as attribute threadCounts in the test result document **Possible values:** • *yes*: Output data type • *no*: Do not output any data type (default)
filename	Output file name as filename attribute in test result document **Possible values:** • *yes*: Output file name • *no*: Do not output a file name
assertionResults	Output result as assertionResults field in test result document (only if outputFormat parameter is set to *CSV*) **Possible values:** • *none*: Do not output a result (default) • *first*: Output first result • *all*: Output all results
printFieldNames	Define the field names in the first line of the test result document (only if the outputFormat parameter is set to *CSV*). **Possible values:** • *yes*: Output field names in the first line • *no*: Do not output field names in the first line (default)
defaultDelimiter	Separator between the individual data fields of a record in the test result document (only if parameter `outputFormat` is set to *CSV*). **Possible values:** • Any character except escape sequences like `\t` • `,`: comma (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.18.2 JMX MBean

? Unknown Attachment `JMX MBean`: Enables access to MBeans of the local JMX server. MBeans can be searched, MBean attributes can be read and set, MBean operations can be called and MBean metadata and key properties can be output separately.
This adapter expects a specific input XML structure depending on the selected operation. The adapter always outputs an XML document that contains information about MBeans, the results of the called MBean methods or, if applicable, error messages, depending on the selected operation.

Former name: *JMX MBean Manipulation Adapter*

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *`List MBeans`*: List all available *MBeans* with name, domain and key properties (no input XML document required). • *`Find MBeans`*: Search MBean by object name; expects a specific input XML document; see operation `Find MBeans` • *`Execute Operations`*: Execute JMX operations; expects a specific input XML document depending on the desired operation.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.jmx.JmxAdapter`*: Main class (default)

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

Input

This adapter expects an adapter-specific input XML structure for the operations `Find MBeans` and `Execute Operations`, see below. This input XML structure can be generated dynamically via XSL mappings depending on the processed data.

10.18.2.1 Betrieb: MBeans finden

You can search for MBeans using the `Find MBeans` operation. All MBeans that match the search pattern are output. The adapter expects the following input XML document for this operation.

```
<Elementname>
  <Query objectname="Suchausdruck" attribut="..." />
</Elementname>
```

The input XML document has any root element and any number of `<Query>` elements. Each `<Query>` element contains a search expression for the MBean object name in the attribute `objectname`, which is structured according to the pattern `domainname:key` or `domainname:keyname=value`; see http://java.sun.com/javase/6/docs/api/javax/management/ObjectName.html.

The search expression can contain the wildcards (*?* and ***). All additionally specified attributes are passed through to the output document unchanged.

Note:

- Depending on the application server used, name resolution may be implemented incorrectly. For example, the MBean server from *JBoss* may not resolve placeholders if they are in key names and values.
- Each object name must be qualified by a key. If all instances of the JMX object are to be addressed, add *:** to the object name.

Example

The following search query searches within the domains `de.softproject.X0` to `de.softproject.Xz` for all MBeans whose key (*Key Property*) `name` starts with *X4* and passes an attribute `sampleAttribute="1"` to the output document.

```
<JMX>
  <Query objectname="de.softproject.X?:name=X4*" sampleAttribute="1" />
</JMX>
```

For example, the adapter outputs the following XML document with the search results:

```
<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

The `Execute Operations` operation allows you to call MBean operations, read and set attributes and output MBean metadata and key properties separately. The basic structure of the input XML document is identical to the `Find MBeans` operation, see Operation: Find MBeans, but different elements can be inserted within the `<Query>` element. These elements represent instructions which are explained below.

Note:

- As with Operation `Find MBeans`, with Operation `Execute Operations` the search expression in attribute `objectname` can contain the wildcards *?* and ***.
- Each object name must be qualified by a key. If all instances of the JMX object are to be addressed, add *:** to the object name.

Read metadata

The `<GetMetadata/>` child element of the `<Query>` element instructs the adapter to output all metadata of the corresponding MBeans:

- In the attribute `class`: names the implementation class
- `<Description>`: Contains a description text of the MBean
- `<Constructors>`: Lists constructors with associated operations and description
- `<Attributes>`: Lists all attributes of the MBean
- `<Operations>`: Lists all operations of the MBean with name, description, data types, parameters, etc.
- `<Notifications>`: Lists notifications from the MBean, if applicable.

```
<Query objectname="Suchausdruck" attribut="wert">
   <GetMetadata/>
</Query>
```

Output key properties separately

The child element <ExpandKeyProperties/> of the <Query> element instructs the adapter to output all key properties in a separate <Property> element each:

```
<Query objectname="Suchausdruck" attribut="wert">
   <ExpandKeyProperties/>
</Query>
```

Read out attribute values

The <GetAttributeValue/> child element of the <Query> element instructs the adapter to output values to MBean attributes specified in each <Attribute> child element in the name attribute:

```
<Query objectname="Suchausdruck" attribut="wert">
   <GetAttributeValue>
      <Attribute name="Attribut-Name" />
   </GetAttributeValue>
</Query>
```

Read out attribute values

The <SetAttributeValue/> child element of the <Query> element instructs the adapter to set MBean attribute values. This is done by defining each MBean attribute in an <Attribute> element:

- In the attribute name: Name of the MBean attribute to be set.
- In the attribute class: data type of the MBean attribute to be set.
- The value of the MBean attribute is specified in the element content.

```
<Query objectname="Suchausdruck" attribut="wert">
   <SetAttributeValue>
      <Attribute name="Attribut-Name" class="Datentyp"></Attribute>
   </SetAttributeValue>
</Query>
```

Call MBean operations

The <InvokeOperation/> child element of the <Query> element instructs the adapter to invoke an MBean operation. The <InvokeOperation/> element must contain the name of the MBean operation in the name attribute. In any number of child elements <Parameter> the respective parameter value is specified, in the attribute class the data type of the parameter value is defined.

The following data type specifications in the class attribute can be used:

• *boolean* • *char* • *byte* • *short* • *int* • *long* • *float*	• *double* • *java.lang.Boolean* • *java.lang.Character* • *java.lang.Byte* • *java.lang.Short* • *java.lang.Integer* • *java.lang.Long*	• *java.lang.Float* • *java.lang.Double* • *java.lang.String* • *java.math.BigInteger* • *java.math.BigDecimal* • *java.management.ObjectName*

Since MBean parameters do not have names, the order in which they are noted is decisive.

A call to an MBean operation is thus made by an input XML structure built according to the following pattern:

```
<Query objectname="Suchausdruck" attribut="wert">
   <InvokeOperation name="MBean-Operation">
      <Parameter class="Datentyp">Parameterwert</Parameter>
   </InvokeOperation>
</Query>
```

Example

In the following input XML document, metadata is queried, key properties are listed, MBean attributes are read and set, and an MBean operation with two parameters is executed:

```
<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`: Enables the monitoring of the currently executed Technical Process with a checkpoint that corresponds to a JMX MBean.
If the checkpoint has been passed, the timestamp of the MBean will be updated. The adapter's return status depends on the MBean property `Value` which can have integer values between *0* and *899*. With such a numeric adapter status you can model more complex case distinctions in Technical Processes.

If the corresponding MBean does not exist yet, it will be created. It will receive the current timestamp, and the MBean property `Value` will be set to the parameter value of `Startvalue`. Additionally, the adapter enables you to remove an existing checkpoint MBean from the JMX registry.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • *Check*: Update timestamp and read MBean value `value` • *Unregister*: Remove MBean from the JMX registry

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.jmxtools.JmxNumberAdapter*: Main class (default)
Objectname	Name of the modified object (checkpoint) **Possible values:** • *de.softproject.x4.jmxnumber:pid=%PID%,process=%CURRENT_PROCESS%,userid=%USER_ID%*: Current process instance (default) • Any valid JMX object name, see http://java.sun.com/javase/6/docs/api/javax/management/ObjectName.html
Startvalue	Default value for the MBean property Value **Possible values:** • Any integer number between *0* and *899* • *1*: Set value *1* to output status *1* (default)
Persistent	Persist value of the JMX Bean **Possible values:** • *True*: Persist value • *False*: Do not persist (default)

Input

This function adapter accepts any input document and outputs it without any modification.

Output

The output document is equivalent to the input document.

Status values

0...899	The MBean property Value has a value between *0* and *899*
-1	Error when invoking JMX (e. g. incompatible MBean)

10.18.4 JMX Switch

? Unknown Attachment `JMX Switch`: Enables the monitoring of the currently executed X4 process with a checkpoint that corresponds to a JMX MBean.

If the checkpoint has been passed, the timestamp of the MBean will be updated. The adapter's return status depends on the MBean property `Value`. If the corresponding MBean does not exist yet, it will be created. It will receive the current timestamp, and the MBean property `Value` will be set to the parameter value of `Startvalue`. Additionally, the adapter enables you to remove an existing checkpoint MBean from the JMX registry.

Ehemalige Bezeichnung bis Version 4.5.1: *JMX Switch Adapter*

> JMX Number allows to set an integer value between *0* and *899* instead of a Boolean value. Such a numeric adapter status allows to model more complex case distinctions in X4 processes, see JMX Number.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *Check*: Update timestamp and read MBean value `value` • *Unregister*: Remove MBean from the JMX registry

Parameter

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.jmxtools.JmxSwitchAdapter*: Main class (Default)
Objectname	Name of the modified object (checkpoint) **Possible values:** • *de.softproject.x4.jmxswitch:pid=%PID%,process=%CURRENT_PROCESS%,userid=%USER_ID%*: Current process instance (Default) • Any valid JMX object name, see http://java.sun.com/javase/6/docs/api/javax/management/ObjectName.html
Startvalue	Default value for the MBean property `Value` **Possible values:** • *True*: Yes (Default) • *Nein*: No

Persistent	Persist value of the JMX Bean **Possible values:** • *True*: Persist value • *Nein*: Do not persist (Default)

Input

This function adapter accepts any input document and outputs it without any modification.

Output

The output document is equivalent to the input document.

Status value

0	The MBean property `Value` has value *false*
1	The MBean property `Value` has value *true*
-1	Error when invoking JMX (e. g. incompatible MBean)

10.18.5 SNMP Management

`SNMP Management`: Manages devices in an IP network via the *Simple Network Management Protocol* (SNMP) in version 1, 2c or 3 (with automatically selected security level). The adapter requests single or multiple management datasets (MIB objects) from agents, sets values of management datasets and sends trap messages to agents.
This adapter expects an input XML document with a specific structure containing MIB object IDs and SNMP parameters and outputs an XML document.

> If the agent detects an error while monitoring the system, it can report this to the *SNMP Management Adapter* without being asked by means of a trap packet. These packets are not confirmed by the adapter. The agent cannot therefore determine whether the trap has arrived at the adapter.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *GET*: Request individual management data set (MIB object). The agent responds with a response packet containing either the requested data or an error message. • *GETNEXT*: Request subsequent management record (MIB object) (to traverse tables) • *GETBULK*: Request several management data sets (MIB objects) simultaneously • *SET*: Change one or more management data sets (MIB objects) of a unit or trigger actions. The agent confirms the acceptance of the values with a response packet. • *SENDTRAP*: Send trap message about an event unsolicited to the manager via UDP from port 161.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.snmp.SnmpAdapter*: Main class (default)
host	IP address or host name of the network management station (NMS) **Possible values:** Any valid string (e.g. *127.0.0.1*)
port	UDP port number (can only be changed for SNMP in version 2c and 3) **Possible values:** • *161*: Default port on which SNMP agents listen for the operations *GET*, *GETNEXT*, *GETBULK* and *SET* (default). • *162*: Port on which the network management station listens by default (for the *SENDTRAP* operation).

version	SNMP version used **Possible values:** • *`version1`*: Use version 1 • *`version2c`*: Use version 2c • *`version3`*: Use version 3
timeout	Time limit for establishing connection in milliseconds **Possible values:** • Any integer • *`3`*: Time limit of 3 ms (standard)
retries	Number of connection retries if the connection cannot be established directly **Possible values:** • Any integer • *`0`*: No additional connection attempt (default)
walk	Execute all queries in the input XML document with the *Walk* operation until all instances in the OID subtree have been queried (if the Operation property is set to *`GETNEXT`* or *`IGETBULK`*) **Possible values:** • *`true`*: Execute *`GETNEXT`* and *`GETBULK`* requests with the *SNMP Walk* operation • *`false`*: Do not execute a *`Walk`* operation unless otherwise defined in the input XML document (default).
contextEngineId	SNMP entity to execute the request (if `version` parameter is set to *`version3`*) **Possible values:** Any string
community	Name of the community of the SNMP agents (if the parameter version is set to *`version1`* or *`version2c`*) **Possible values:** Any string (e.g. *`public`*)

securityname	Security name for authentication of messages via SNMP version 3 (if parameter `version` is set to *`version3`*) **Possible values:** any string
authenticationprotocol	Method used to authenticate a message via SNMP version 3 (if parameter `version` is set to *`version3`*) **Possible values:** • *`MD5`*: Create *MD5* checksum • *`SHA`*: Create *Secure Hash* Code
authenticationpass	Password for authentication via SNMP version 3 on security level *authNoPriv* (if parameter `version` is set to *`version3`*) **Possible values:** Any string with at least 8 characters length
privacyprotocol	Encryption algorithm for messages with SNMP version 3 (if parameter `version` is set to *`version3`*) **Possible values:** • *`DES`*: Use *Data Encryption Standard* algorithm • *`AES`*: Use *Advanced Encryption Standard* algorithm • *`AES192`*: Use *Advanced Encryption Standard* algorithm with 192 bit key length • *`AES256`*: Use *Advanced Encryption Standard* algorithm 256 bit key length
privacypass	Password for encrypting messages with SNMP version 3 at security level `authPriv` (if parameter `version` is set to *`version3`*) **Possible values:** Any string

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Input

This adapter expects an adapter-specific input XML structure, where each `<pdu/>` element stands for a network device.

```
<root>
    <pdu oid="MIB-Objekt-ID" timeout="Zeitbeschränkung in ms"
         retries="Anzahl zusätzlicher Verbindungsversuche" />
</root>
```

Output

This adapter outputs a UTF-8 encoded XML document with the following structure.

```
<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` Reads the event log of Microsoft Windows and outputs all event entries or those selected according to filter criteria in an XML document.
Former name: *Windows Event Logging Adapter*

ⓘ The following conditions must be met in order to use this adapter:

- The Windows computer from which the event log is to be read must be in the same domain as the *X4 Server*.
- The *X4 Server* must be running on a Windows computer that has domain administrator rights.
- A JNI licence is required.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Execute`*: Execute adapter

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.windowseventlog.WindowsEventLogAdapter`*: Main class (default)

log	Windows log/log whose events are to be displayed **Possible values:** • Any Windows protocol/log (e.g. *`Application`*) • *`System`*: Output system events (default)
from	Start date to restrict to a time period **Possible values:** Any date and time in the format `YYYY-MM-DDThh:mm:ss` (e.g. *`2012-03-31T13:45:03`*)
to	End date to limit to a period **Possible values:** Any date and time in the format `YYYY-MM-DDThh:mm:ss` (e.g. *2012-03-31T13:45:03*)
type	Filter by an event type (level) **Possible values:** Any event type (e.g. *`Error`*)
source	Filter by an event source (this can be an application, for example) **Possible values:** Any string (e.g. *`PrintSpooler`*)
category	Filter by event category **Possible values:** Any character string
eventId	Filter by an Event ID **Possible values:** Any integer
user	Filter by a user **Possible values:** Any string (e.g. *`SYSTEM`*)
computer	Name of the computer from which the event log is to be read **Possible values:** • Any computer name or IP address • *`127.0.0.1`*: Currently used system, i.e. localhost (default)

Status values

1	The operation was executed successfully.

`-1`	The operation failed due to a technical error.

Output

This function adapter outputs a UTF-8 encoded XML document with the following structure. For each event, an XML element `<log/>` is created that contains a set of information as well as the event description for the event:

```
<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`: Generates a list of Windows services and their status and outputs them in an XML document.

ⓘ The following requirements must be ensured in order to use this adapter:

- The Windows computer from which service status information is to be read must be in the same domain as the *X4 Server*.
- The *X4 Server* must be running on a Windows computer that has domain administrator rights.
- A JNI licence is required.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Execute`*: Execute adapter

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.windowsservice.WindowsServiceAdapter`*: Main class (default)
host	Name of the computer executing the Windows service **Possible values:** • Any computer name or IP address • *`127.0.0.1`*: Currently used system, i.e. localhost (default)

service	Filter by a specific Windows service **Possible values:** Any character string

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Output

This adapter outputs a UTF-8 encoded XML document with the following structure. For each service, an XML element `<service/>` is created that contains information about the Windows service as well as its status:

```
<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`: Establishes a connection to websms and sends SMS.

To establish a connection with websms and send SMS, a websms user account is required. Additionally, the two-factor authentication must be activated.

For more information, visit the websms website.

Parameters

`Adapter`	Adapter main class (do not change!) **Possible values:** • *`de.softproject.x4.adapter.checkrights.CheckRightsAdapter`*: Main class (default)
`targetPhoneNumbers`	Phone numbers of the recipients in MSISDN format. Multiple phone numbers are separated by commas.
`websmsUsername`	Username of the websms user account

`websmsPassword`	Password of the websms user account
`websmsAccessToken`	API token ⚠ The API token must be created in the API management of websms.
`deliveryReportCallbackURL`	URL for forwarding delivery reports. ⓘ (websms API: notificationCallbackUrl)
`simulation`	Simulation mode: If the simulation mode is activated, no SMS will be sent. ⓘ (websms API: test)

Status values

`1`	The adapter operation was executed successfully.
`-1`	An error occurred during the execution of the adapter.

Input

The adapter expects an XML document as input. Only the content of the root element is sent as SMS.

```
<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`: Converts a text document containing structured data in J*avaScript Object Notation (JSON)*to an XML document, and conversely converts an XML document to a UTF-8 encoded

JSON document. In doing so, the adapter uses a new XML format that allows better control of the generated JSON document.

This adapter replaces the deprecated JSON Converter.

Properties

`Operation`	Determines which operation the adapter executes **Possible values:** • `JSON2XML`: Convert JSON document to XML • `XML2JSON`: Convert XML document to JSON

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.json.Json2Converter`*: Main class (default)
`allowBareObjectKeys`	Information on how to handle Object Keys that are not enclosed in quotation marks **Possible values:** • `true`: Allows Object Keys that are not enclosed in quotes • `false`: does not allow Object Keys that are not enclosed in quotes
`escapeStringInXml`	Information about handling non-XML characters **Possible values:** • `true`: Escape non-XML characters. Characters which are forbidden in XML are replaced by `\u????` where the question marks correspond to the hexadecimal digits of the Unicode character • `false`: Filter non-XML characters. Characters that are forbidden in XML are replaced by the Unicode character `uFFFD` (Replacement Character).

`emptyStringAsNull`	Output empty strings as `null` for numbers and Boolean values **Possible values:** • `true`: output empty strings as `null`, e.g. *`<Value type="Boolean"></Value>`* is output as null • `false`: Do not output empty strings as `null` , e.g. *`<Value type="Boolean"></Value>`* generates an error. `<Value type="String"></Value>` always generates `""`.

The `XML2JSON` operation ignores whitespaces at the beginning and end of the value for numeric and boolean values.

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Example

The following examples show JSON expressions and their respective equivalents 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>`

JSON	XML
`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/>`
`{ "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>`

JSON	XML

{
"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`: Validates an input JSON document against a JSON schema. If the verified JSON document is valid, the adapter will output the document unalteredly. Otherwise, an XML document with a corresponding error message will be output.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** *`Validate`*: Validate the JSON document

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.json.JSONValidator*: Main class (Default)
schemaURL	Path to the schema definition (JSON schema) **Possible values:** Any `xstore://` URL (e. g. *`xstore://Project/Folder/Schema.json`*)

Status values

-1(error)	An error occurred during the adapter's execution. You can find more information in the server log.
1(successful)	The verified JSON document is valid according to the defined schema. The adapter outputs the verified input JSON document unalteredly.
0	The verified JSON document is not valid according to the defined schema. The adapter outputs a corresponding error message (see Output).

Input

The adapter expects a JSON document as input, which is subsequently validated against a JSON schema.

Output

In case of a successful validation, the adapter outputs the input document unalteredly. Otherwise, if the JSON document ist not valid, an XML document with a corresponding error message is output. In this case, for each error an XML element `<Error/>` is generated, which contains a detailed error description, e.g.:

Example of an error message

```
<ValidationErrors>
    <Error>#/rectangle/a: -5.0 is not higher or equal to 0</Error>
</ValidationErrors>
```

10.20.2 OData Connector

`OData Connector`: Enables communication with systems that provide ReST services according to the OData standard, e.g. SAP S4/Hana, Microsoft Sharepoint, Microsoft Dynamics Navision and others.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `ReadMetadata`: Retrieve the data model and operations exposed by a specific service. • `ReadCollections`: Retrieve all collections available from the service. • `ReadEntitySet`: Retrieve a particular collection from the service. • `ReadSingleEntity`: Retrieve a single entity from the service specified by its collection and ID. • `ReadNavigationEntity`: Retrieve the related collection to the entity. • `ReadAssociation`: Retrieve the related association to the entity. • `ReadSingleton`: Retrieve the specific singleton. • `ReadMediaEntity`: Retrieve the media entity. • `CreateEntity`: Create new entities in the system, which are specified by Collection and ID in an input XML document. • `UpdateEntity`: Update specific entities existing in the system, specified by Collection and ID in an input XML document. • `DeleteEntity`: Delete specific entities existing in the system, specified by Collection and ID in an input XML document. • `CreateSingleton`: Create a new singleton in the system, specified by Collection and ID in an input XML document. • `UpdateSingleton`: Update a specific singleton existing in the system, specified by Collection and ID in an input XML document. • `DeleteSingleton`: Delete a specific singleton existing in the system, specified by Collection and ID in an input XML document. • `CreateMediaEntity`: Create new media entities in the system, specified by Collection and ID within the parameters `entitySet` and `entityId`. • `UpdateMediaEntity`: Update specific media entities existing in the system, specified by Collection and ID within the parameters `entitySet` and `entityId`. • `DeleteMediaEntity`: Delete specific media entities existing in the system, specified by Collection and ID within the parameters `entitySet` and `entityId`. • `Functions`: Request a specific operation (Function, Action) exposed by the service, retrieving correspondent output values.

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.adapter.odataquery.ODataQueryAdapter*: Main class (default)
`serviceUrl`	OData service URL, e.g. `https://services.odata.org/V4/TripPinServiceRW`
`entitySet`	Entity collection to interact with.
`entityId`	Entity ID to interact with.

`navigationProperty`	Retrieve related collection to the entity
`association`	Retrieve related association to the entity
`singletonId`	Singleton ID
`contentTypeMetadata`	ContenType of the service metadata **Possible values:** • `Full`: Output complete metadata • `Minimnal`: Output only basic metadata • `None`: Do not output any metadata

Status values

`1`	The operation was executed successfully and an output is generated
`-1`	An error occurred during the operation's execution an no output is generated.

Input

Depending on the selected operation, the adapter expects the following input structure:

- The operations `ReadaMetadata`, `ReadSingleEntity`, `ReadNavigationEntity`, `ReadSet` and `ReadSingleton` do not require any input document. However the parameters `entitySet` and `entityId` are required.
- The operation `ReadAssociation` does not require any input document. However the parameters `entitySet`, `entityId` and `association` are required.
- Operation `ReadEntitySet`:
  The following system query options can be used with the operation `ReadEntitySet`:
  - `$count`: Ignores any `$top`, `$skip`, or `$expand` query options, and returns the total count of results across all pages including only those results matching any specified `$filter` a nd `$search`.
  - `$skip`: If `$top` and `$skip` are used together, `$skip` **must** be applied before `$top`, regardless of the order in which they appear in the request.
  - `$top`: Allows the user of an OData service to specify the maximum number of entries that should be returned, starting from the beginning.
  - `$orderby`: When requesting a list of entities from a service, it is up to the service implementation to decide in which order they are presented.
  - `$filter`: When requesting a list of entities from a service, the default behaviour is to return all entities on the list. The consumer of an OData service might want to be able to receive a subset by specifying certain criteria which each of the returned entities have to fulfill.
  - `$select`: When requesting an entity collection from the backend, the OData service returns a list of entities and each entity contains a list of properties.
    - Specify one property name only: `$select=Name`
    - Specify a comma-separated list of properties: `$select=Name,Description`
    - Specify a star to include all properties: `$select=*`

- `$expand`: Indicates the related entities and stream values that **must** be specified inline.
- `$search`: Restricts the result to include only those items matching the specified search expression.
- `$format`: Specifies the media type of the response.

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>
```

- Operations `CreateEntity`, `UpdateEntity` and `DeleteEntity`:

CreateEntity, UpdateEntity and DeleteEntity Request

```
<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>
```

- Operations `CreateSingleton`, `UpdateSingleton` and `DeleteSingleton`:

CreateSingleton, UpdateSingleton and DeleteSingleton Request

```
<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>
```

- For the operations `CreateMediaEntity`, `UpdateMediaEntity` and `DeleteMediaEntity` the parameters `entitySet` and `entityId` are required as well as the media input within the adapter.
- Operation `ExecuteFunction`:

ExecuteFunction Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
       <Function name="CountCategories">
             <Parameter name="Amount" value="2"></Parameter>
       </Function>
</Request>
```

Supported types

Adapter Input Type	Type	Meaning
`Binary`	`Edm.Binary`	Binary data
`Boolean`	`Edm.Boolean`	Binary-valued logic
`Byte`	`Edm.Byte`	Unsigned 8-bit integer
`Date`	`Edm.Date`	Date without a time-zone offset

Adapter Input Type	Type	Meaning
`DateTimeOffset`	`Edm.DateTimeOffset`	Date and time with a time-zone offset, no leap seconds
`Decimal`	`Edm.Decimal`	Numeric values with decimal representation
`Double`	`Edm.Double`	IEEE 754 binary64 floating-point number (15-17 decimal digits)
`Duration`	`Edm.Duration`	Signed duration in days, hours, minutes, and (sub)seconds
`Guid`	`Edm.Guid`	16-byte (128-bit) unique identifier
`Int16`	`Edm.Int16`	Signed 16-bit integer
`Int32`	`Edm.Int32`	Signed 32-bit integer
`Int64`	`Edm.Int64`	Signed 64-bit integer
`SByte`	`Edm.SByte`	Signed 8-bit integer
`Single`	`Edm.Single`	IEEE 754 binary32 floating-point number (6-9 decimal digits)
`String`	`Edm.String`	Sequence of UTF-8 characters

Output

Depending on the selected operation, the adapter outputs the following structures:

- Operation `ReadMetadata`:

ReadMetadata Response

```
<?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>
```

- Operations `ReadSingleEntity` and `ReadSingleton`:

ReadSingleEntity Response

```
<?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`:

ReadEntitySet Response

```
<?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`:

ReadCollections Response

```
<?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>
```

- Operations `CreateEntity`, `UpdateEntity` and `DeleteEntity`:

CreateEntity, UpdateEntity and DeleteEntity Response

```
<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>
```

- Operations `CreateSingleton`, `UpdateSingleton` and `DeleteSingleton`:

CreateSingleton, UpdateSingleton and DeleteSingleton Response

```
<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`:

ExecuteFunction Response

```
<?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`: Converts a URL or a string into a URL or vice versa.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `Encode`: Encrypts the specified URL • `Decode`: Decodes the specified URL • `EncodePlain`: Encrypts the specified text • `DecodePlain`: Decodes the specified text

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** *en.softproject.integration.adapter.url.UrlEncodingConverter*: Main class (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation could not be executed.

Input

The adapter expects an XML file with the string or URL to be converted as input.

- Operation `Encode`:

  Expected structure for the operation Encode

  ```
  <?xml version="1.0" encoding="UTF-8" ?>
  <URL>http://www.softproject.de/search?q=Hellö Wörld@X4$</URL>
  ```

- Operation `Decode`:

  Expected structure for the operation Decode

  ```
  <?xml version="1.0" encoding="UTF-8" ?>
  <URL>http://www.softproject.de/search?q=Hell%C3%B6%20W%C3%B6rld@X4$</URL
  ```

- Operation `EncodePlain`:

  Expected structure for the operation EncodePlain

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <Text>Hellö Wörld@X4$</Text>
  ```

- Operation `DecodePlain`:

Expected structure for the operation DecodePlain

```
<?xml version="1.0" encoding="UTF-8"?>
<Text>Hell%C3%B6%20W%C3%B6rld@X4$</Text>
```

Output:

The adapter outputs an XML document with the converted URL.

- Operation `Encode`

Example output for the operation Encode

```
<?xml version="1.0" encoding="UTF-8" ?>
<URL>http://www.softproject.de/search?q=Hell%C3%B6%20W%C3%B6rld@X4$</URL
```

- Operation `Decode`

Example output for the Decode operation

```
<?xml version="1.0" encoding="UTF-8" ?>
<URL>http://www.softproject.de/search?q=Hellö Wörld@X4$</URL>
```

- Operation `EncodePlain`:

Example output for the operation EncodePlain

```
<?xml version="1.0" encoding="UTF-8"?>
<Text>Hell%C3%B6%20W%C3%B6rld@X4$</Text>
```

- Operation `DecodePlain`:

Example output for the operation DecodePlain

```
<?xml version="1.0" encoding="UTF-8"?>
<Text>Hellö Wörld@X4$</Text>
```

10.20.4 URL Reader

`URL Reader`: Reads a document from a URL (*Uniform Resource Locator*) and outputs this document.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *Read*: Read document from a URL

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.urlreader.URLReader*: Main class (default)
Url	Uniform Resource Locator (URL) with any valid protocol (`http://`, `ftp://`, `xstore://`, `file://` etc.) **Possible values:** Any valid URL (e.g. *https://www.google.com/favicon.ico*)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.20.5 Webservice

10.20.5.1 HTTP(S) Connector

`HTTP(S) Connector`: Communicates with servers using the *HyperText Transfer Protocol (HTTP)* or *HTTP Secure (HTTPS)* based on the *Apache HttpClient* in version 4. This adapter provides the standard HTTP methods `GET`, `POST`, `PUT`, `DELETE`, `HEAD`, `OPTIONS`, `TRACE` and `PATCH`. Moreover, user-defined HTTP methods can be used. It supports HTTP basic, digest and NTLMv2 authentication, as well as proxy servers and preemptive authentication. Moreover, the adapter supports Server Name Indication for HTTPS connections, and client certificates to validate requests.

Former name until version 4.5.1: *HTTP(S) Client 4 Adapter*

Please note:

- In default mode (parameter `fromXML` is disabled), the adapter processes, depending on the operation, any input document using its original content type and encoding.
- In expert mode (parameter `fromXML` is enabled), the adapter processes only a specific XML document that may include headers, cookie information etc. and which allows to have more control of the HTTP(S) operation, see parameter `fromXML` and input description below..
- This adapter replaces the existing adapter *HTTP(S) Communication*, see HTTP(S) Communication (deprecated).

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *Get*: Retrieve data from a URL using the method `GET` • *Post*: Send data in the body of the HTTP request to a URL using the `POST` method • *Put*: Publish data in the body of the HTTP request to a URL using the `PUT` method • *Delete*: Delete the resource specified in the HTTP request using the `DELETE` method • *Head*: Retrieve header information using the `HEAD` method • *Options*: Retrieve a list of supported HTTP methods from the server using the `OPTIONS` method • *Trace*: Return the request received by the server, so that the client can see what changes have been made by intermediate servers • *Patch*: Partially update the data deposited within the HTTP request's body • *Execute*: Send data within the HTTP request's body with any HTTP method, see also Input example

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.http4.HttpClient4Adapter*: Main class (Default)
protocol	Used protocol (HTTP or HTTPS) **Possible values:** • `http`: Use HTTP (no encryption) (Default) • `https`: Use HTTPS (SSL-encrypted)

host	IP address or host name of the physical server **Note:** The `host` parameter adds an `HTTP header` named `host`. **Possible values:** Any valid string without protocol handler (e. g. *`google.com`*)
vHost	Virtual host name, if it cannot be identified by the physical host within the parameter `host` **Note:** The `vHost` parameter overrides the value for the `HTTP header` named `host`, see `host` parameter. The ports can be specified, except for ports 80 (default for HTTP connections) and 443 (default for HTTPS connections). **Possible values:** Any valid string without protocol handler (z. B. *`google.com`*)
port	Port number of the HTTP(S) service **Possible values:** • Any integer number (e. g. *`8080`*) • `-1`: Use port 80 for HTTP and port 443 for HTTPS (Default)
path	Path on the server, starting with a slash **Possible values:** Any valid path (e. g. */`intl/en_ALL/images/logo.gif`*)
method	Name of the HTTP method to be used for the operation `Execute`. Optionally, the method can also be set via the XML request. **Possible values:** Any string

authenticationEncoding	Character encoding for the credentials **Possible values:** Specification of a character encoding, e.g. `UTF-8` oder `ISO-8859-1` ℹ With the help of this parameter, umlauts can also be used in the credentials.
query	Argument(s) of the HTTP(S) request **Possible values:** Any URL-encoded arguments without leading question marks (e. g. *`Arg1=V1&Arg2=V2`*)
username	User name for basic HTTP, digest, or NTLM authentication **Possible values:** • For HTTP Basic authentication: the user name • For HTTP Basic authentication on a Windows domain: a string following the pattern `<Domain>\<User name>` (e. g. *`MyDomain\MyUserName`*) • For NTLM authentication on a Windows domain: only the user name (and specify the domain in parameter `ntlmDomain`)(e. g. *`MyUsername`*)
password	Password for basic, digest, or NTLM authentication **Possible values:** Any valid string
ntlmWorkstation	Workstation (computer name) of the client (only for NTLM authentication, see parameter `allowNtlm`) **Possible values**: Any valid workstation name
ntlmDomain	Domain name (only for NTLM authentication, see parameter `allowNtlm`) **Possible values:** Any valid domain name (NetBIOS name)

usePreemptiveBasicAuth	Use preemptive authentication (using credentials from the parameters `username` and `password`) for all HTTP(S) requests (only for basic authentication, see parameter `allowBasic`) **Possible values:** • `true`: Send credentials directly (preemptively) with each HTTP(S) request • `false`: No preemptive authentication (Default)
allowBasic	Use HTTP(S) basic authentication **Possible values:** • `true`: Use basic authentication • `false`: Do not use basic authentication (Default)
allowDigest	Use digest authentication **Possible values:** • `true`: Authenticate with digest • `false`: Do not authenticate with digest (Default)
allowNtlm	Use proprietary NTLM server authentication by Microsoft **Possible values:** • `true`: Authenticate with NTLM • `false`: Do not authenticate with NTLM (Default)

fromXml	Enable expert mode; The adapter now expects a specific XML document including headers, cookie information etc. that allows more control of the HTTP(S) operation **Note:** In expert mode, the `HTTP header host` can be set using the `host` parameter only (not in the XML document). Alternatively, the `host` parameter can be overridden by the `vHost` parameter, see the `host` or `vHost` parameters. **Possible values:** • `true`: Enable expert mode (adapter expects a specific XML document) • `false`: Process adapter input directly
toXml	Output an XML document containing all headers as elements and the body content as Base64-encoded string **Possible values:** • `true`: Output an XML document • `false`: Output the unmodified response of the server (Default)
hideCookieHeader	Output header elements also with cookie information in the result XML document (when parameter `toXml` is enabled); Disable only for development/test or for non-standard cookies. **Possible values:** • `true`: Do not output cookie information as header in the result XML document (Default) • `false`: Output cookie information as header in the result XML document
useProxy	Use a proxy server to connect **Possible values:** • `true`: Connect via the proxy server, specified within the parameter `proxyHost` • `false`: No proxy server (Default)

proxyHost	IP address or host name of a proxy server **Possible values:** Any valid string (e. g. *`192.168.1.1`*)
proxyPort	Port number of the proxy server **Possible values:** • Any integer number • `-1`: Use default port (Default)
proxyUsername	User name for Basic HTTP, digest, or NTLM authentication on a proxy server **Possible values:** • For HTTP Basic authentication: the user name • For HTTP Basic authentication on a Windows domain: a string following the pattern `<Domain>\<User name>` (e. g. *`MyDomain\MyUserName`*) • For NTLM authentication on a Windows domain: only the user name (and specify the domain in parameter `ntlmDomain`) (e. g. *`MyUsername`*)
proxyPassword	Password of a proxy server **Possible values:** Any valid string
proxyNtlmWorkstation	Workstation (computer name) where the authentication request comes from (for an NTLM authentication via NTLM proxy) **Possible values:** Any valid workstation name
proxyNtlmDomain	Windows domain for an NTLM authentication via NTLM proxy **Possible values:** Any valid domain name

proxyUsePreemptiveBasicAuth	Enable preemptive authentication (using credentials from the parameters `proxyUsername` and `proxyPassword`) for the proxy server (only for basic authentication, see parameter `proxyAllowBasic`) **Possible values:** • `true`: Send credentials directly (preemptively) with each proxy request • `false`: No preemptive authentication (Default)
proxyAllowBasic	Use basic authentication for the proxy **Possible values:** • `true`: Use basic authentication for the proxy • `false`: Do not use basic authentication for the proxy (Default)
proxyAllowDigest	Use digest authentication for the proxy **Possible values:** • `true`: Authenticate proxy with digest • `false`: Do not authenticate proxy with digest (Default)
proxyAllowNtlm	Use proprietary NTLM server authentication by Microsoft for the proxy **Possible values:** • `true`: Authenticate proxy with NTLM • `false`: Do not authenticate the proxy with NTLM (Default)
connectTimeout	Timeout value for a socket connect in milliseconds **Possible values:** • `>0`: Timeout value in milliseconds • `<=0`: No timeout (default)

socketTimeout	Timeout value between two incoming data packages in milliseconds. This is not the maximal request time, only the maximal idle time of the connection. **Possible values:** • `>0`: Timeout value in milliseconds • `<=0`: No timeout (default)
redirectsEnabled	Allow URL redirects for server responses with HTTP status *`301 Moved Permanently`* **Possible values:** • `true`: URL-Redirects ausführen (default) • `false`: Keine URL-Redirects ausführen
circularRedirectsEnabled	Allow circular URL redirects for server responses with HTTP status *`301 Moved Permanently`* **Possible values:** • `true`: Follow circular redirects (default) • `false`: Do not follow circular redirects
maxRedirects	Maximum number of URL redirects for server responses with HTTP status *`301 Moved Permanently`* **Possible values:** • Any integer number • *`-1`*: No limitation (default)
acceptAllCertificates	Disable certificate checking **Possible values:** • `any string`: Certificates are checked on confidence (default) • `Yes, I want to disable security by accepting untrusted certificates.`: Certificate check is disabled **Please note:** The exact string must be entered here!

acceptAllSslHostnames	Disable hostname checking for certificates **Possible values:** • `any string`: Hostname must correspond to the certificate (default) • `Yes, I want to disable security by accepting non-matching hostnames.`: Hostname checking is disabled **Please note:** The exact string must be entered here!
useClientCertificate	Use client certificates to validate the request **Possible values:** • `true`: Use client certificates to validate the request • `false`: Do not use client certificates to validate the request (default)
keystoreType	Keystore type for storing client certificates (see http://docs.oracle.com/javase/6/docs/technotes/guides/security/StandardNames.html#KeyStore) **Possible values:** • `jceks`: Use proprietary keystore implementation by SunJCE • `jks`: Use proprietary keystore implementation by SUN • `pkcs12`: Use keystore transfer format PKCS12
keystoreURL	Keystore URL to validate the request via client certificates **Possible values:** Any valid URL, e.g. `file:/C:/Folder/ExampleKeystore`
keystorePassword	Password for the keystore to validate the request via client certificates **Possible values:** Any valid string
keyAlias	Key alias to access a specific key within the keystore **Possible values:** Any valid string
aliasPassword	Corresponding password for a specific key within the keystore **Possible values:** Any valid string

State values

-1	Technical error: Server does not respond, or the adapter input XML cannot be parsed (only if expert mode has been enabled via the parameter `fromXml`)
>=100	Three-digit HTTP status code depending on the response, see http://tools.ietf.org/html/rfc2616#section-6.1.1

Input (*Default mode*)

By default (parameter `fromXML` is disabled), depending on the operation, this adapter processes any input document using its original content type and encoding.

Input (*Expert mode*)

In expert mode (Parameter `fromXML` enabled) the adapter processes a specific XML document that may include headers, cookie information etc. that allows more control of the HTTP(S) operation.

This XML structure corresponds to the adapter's output XML structure, if the parameter `toXml` is enabled. In addition, it almost completely corresponds to the *X4 ReST* interface (If in the configuration file `<X4>/X4DB/0/restconfig.xml` an attribute *`isDirect="false"`* has been defined for a URI operation). The main difference to the *X4 ReST* result XML document is the name of the element that contains the Base64-encoded body (`<Content>` instead of `<Body>` for the adapter).

Within the element `<Cookies>`, cookies can be specified, each within in an element `<Cookie>`:

```
<Request>
   <Headers>
      <Header>
         <Name><!--HTTP header name--></Name>
         <Value><!--Header value--></Value>
      </Header>
      <!-- Or for date values: -->
      <Header>
         <Name><!--HTTP header name--></Name>
         <Date timezone="Timezone"><!--Header date value--></Date>
      </Header>
   </Headers>
   <Cookies>
      <Cookie expires="Expiry date" path="Path" domain="Domain">
         <Version><!--Version number--></Version>
         <Secure/>       <!-- If Secure attribute is set -->
         <Persistent />  <!-- For a persistent cookie -->
         <Domain><!--Domain--></Domain>
         <Path><!--Path--></Path>
         <Name><!--Cookie name--></Name>
         <Value><!--Cookie value--></Value>
         <ExpiryDate><!--Expiry date in format YYYY-MM-DD hh:mm:ss--></ExpiryDate>
      </Cookie>
   </Cookies>
   <Body><!--Base64-encoded data of the HTTP request--></Body>
</Request>
```

Operation Execute

For the operation `Execute` in conjunction with the activated parameter *`fromXML`*, the HTTP method can also be set within the Input XML as follows:

```
 <Request method="CUSTOM_METHOD">
   <Headers>
      <!-- ... -->
   </Headers>
   <Cookies>
      <!-- ... -->
   </Cookies>
   <Body><!--Base64-encoded data of the HTTP request--></Body>
</Request>
```

Output

By default (if parameter `toXml` is disabled) the result document of the HTTP(S) request will be outputted.

If the parameter `toXml` is enabled, an XML document will be outputted, that contains all HTTP header information of the response within element `<Header>`. If a header value contains multiple values

separated by comma, these parts will be also outputtes each in an element `<Element>` within the children elements `<Name>` and `<Value>`.

The body content will be outputted as Base64-encoded string in the element `<Body>`. If the parameter `hideCookieHeader` has been disabled, each header causing the cookie will be outputted additionally.

```
<Response>
   <Request line="HTTP request line">
      <Method><!--HTTP method--></Method>
      <URI><!--Path on the server--></URI>
      <Protocol>HTTP/1.1</Protocol>
   </Request>
   <Status line="HTTP status line">
      <Code><!--Three-digit status code--></Code>
      <Reason><!--Status name--></Reason>
      <Protocol><!--HTTP/1.1</Protocol>
   </Status>
   <Headers>
      <Header>
         <Name><!--HTTP header name--></Name>
         <Value><!--Header value--></Value>
      </Header>
      <!-- Or in case of date values: -->
      <Header>
         <Name><!--HTTP header name--></Name>
         <Date timezone="Timezone"><!--Header date value--></Date>
      </Header>
   </Headers>
   <Cookies>
      <Cookie expires="Expiry date" path="Path" domain="Domain">
         <Version><!--Version number--></Version>
         <Secure/>       <!-- If attribute secure is set -->
         <Persistent />  <!-- For a persistent cookie -->
         <Domain><!--Domain--></Domain>
         <Path><!--Path</Path>
         <Name><!--Cookie name--></Name>
         <Value><!--Cookie value--></Value>
         <ExpiryDate><!--Expity date in the format YYYY-MM-DD hh:mm:ss--></
ExpiryDate>
      </Cookie>
   </Cookies>
   <Body><!--Base64-encoded data of the response body--></Body>
</Response>
```

10.20.5.2 Webservice (SOAP) Connector

> **Note**
> The `Webservice (SOAP) Connector` will be deprecated with Release 8.0 of the X4 BPMS and no longer be supported in future releases.

`Webservice (SOAP) Connector`: Sends a *SOAP Request* via HTTP(S) to the URL of a web service and issues the returned *SOAP Response*. It communicates with servers via HTTP (Hyper Text Transfer Protocol) or HTTPS (HTTP Secure).

If authentication is required, this must be SOAP-based via the *SOAP header*; its structure (certificate-based, token-based, etc.) must first be clarified with the remote peer. HTTP and NTLM authentication and proxy servers are also supported.

Former name: *HTTP WebService Request Adapter*

> The adapter supports *SOAP MTOM* (*Message Transmission Optimization Mechanism*), which can be used to transmit large documents or binary data in MTOM-based WebServices via XOP (XML-binary Optimized Packaging) with high performance, see WebService Starter (veraltet). MTOM processing for requests is enabled only if the adapter's input document has the MIME type `text/xml` or `application/xml`.
> The MTOM function is controlled via Processing Instructions, which are added to the SOAP request document to enable MTOM on input side:
>
> - `<?useMTOM?>`: enable MTOM function for this SOAP request
> - `<?MTOM-xpath 0-n XPath expressions ?>`: Any number of XPath expressions to define the elements containing binary data
> - `<?MTOM-size size in KB ?>`: size of the data in kilobytes from which compression is activated (default: 1 KB)
>
> In addition, the adapter automatically converts MTOM-based SOAP responses provided by a web service into standard SOAP responses.

Properties

Operation	Determines which operation the adapter executes **Possible values:** `Call`: Execute adapter

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.httpws.HttpWSAdapter*: Main class (default)
wsURL	URL to which the SOAP request is sent, with the default port *80*. **Possible values:** Any web service URL Deviating ports must be specified in the URL.
soapAction	HTTP header of the SOAP request. The value of this parameter is a URL that identifies the requested operation.
login	Username for simple HTTP authentication **Possible values:** Any string
pwd	Password for simple HTTP authentication **Possible values:** Any string
preemptiveAuth	Enable preemptive authentication (using credentials of `login` and `pwd` parameters) for all HTTP(S) calls. **Possible values:** • *Yes*: Send credentials directly (preemptively) with each HTTP(S) call. • *No*: No preemptive authentication (default)
pxHost	IP address or host name of a proxy server **Possible values:** Any valid string (e.g. *192.168.1.1*)
pxPort	Port number of the proxy server **Possible values:** Any integer
pxLogin	Username for the proxy server **Possible values:** Any string
pxPwd	Password for the proxy server **Possible values:** Any string

pxNTLMDomain	Windows domain for NTLM authentication via NTLM proxy **Possible values:** Any domain name
pxNTLMhost	Host (computer name) from which the authentication request originates (for NTLM authentication via NTLM proxy). **Possible values:** Any host name
pxUseNTLM	Use Microsoft's proprietary NTLM authentication for the proxy **Possible values:** • *yes*: Authenticate via NTLM • *no*: Do not authenticate via NTLM (default)
nTLMDomain	Windows domain for NTLM server authentication **Possible values:** Any domain name
nTLMhost	Host (computer name) from which the NTLM server authentication request originates. **Possible values:** Any host name
useNTLM	Use Microsoft proprietary NTLM server authentication **Possible values:** • *yes*: Authenticate via NTLM • *no*: Do not authenticate via NTLM (default)
timeout	Time limit for establishing connection in milliseconds **Possible values:** • Any integer • *0*: No time limit
keystorePath	Complete path to the keystore file; if a keystore is specified, only certificates of this keystore are accepted **Possible values:** • Any path to the keystore file (use only certificates of this keystore) • *(no path)*: Accept all certificates

keystorePassword	Password for the keystore file **Possible values:** Any string

Status values

1	The operation was executed successfully.
-1	The operation failed due to a technical error.

Input

To call a web service with MTOM, the SOAP request must contain the above processing instructions, for example:

```
<?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>
```

Multiple Processing Instructions are always listed as individual elements one after the other.

10.20.5.3 Webservice Security

`Webservice Security`: Encodes, signs and decodes a SOAP message or adds a user token or a timestamp to it. The adapter expects a valid SOAP message as input and outputs a SOAP message modified according to the selected operation.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *`Decode`*: Encode message • *`Decode`*: Decode message • *`Sign Message`*: Sign message • *`Add Username Token`*: Add user name token • *`Add Timestamp`*: Add timestamp

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.wssecurity.WSSecurityAdapter`* : Main class (default)
cryptoProviderClass	Class name for the *WSS4J* crypto provider (provider class must implement the interface `org.apache.ws.security.components.crypto.Crypto`) **Possible values:** • Any class name • *`org.apache.ws.security.components.crypto.Merlin`*: Merlin-Crypto-Provider (default)
keystoreType	Keystore type of the crypto provider **Possible values:** • Any keystore type • *`jks`*: Java Keystore (default)
keystorePassword	Keystore password **Possible values:** Any string
keystoreAlias	Alias name of the keystore **Possible values:** Any string
aliasPassword	Alias password of the keystore **Possible values:** Any string

keystoreFile	Path to the keystore file
encodingAlgorithm	Algorithm for symmetric encryption of a message **Possible values:** • *`Triple DES`*: *Data Encryption Standard* -Use triple encryption algorithm • *`AES 128`*: *Advanced Encryption Standard* -Use algorithm with 128 bit key length • *`AES 128`*: *Advanced Encryption Standard* -Use algorithm with 192 bit key length • *`AES 128`*: *Advanced Encryption Standard* -Use algorithm with 256 bit key length
canonicalization	Canonicalization algorithm for encoding/signing according to http://www.w3.org/TR/xml-exc-c14n/ or http://www.w3.org/TR/xml-c14n
keyEncodingAlgorithm	Symmetric encryption algorithm for the key **Possible values:** • *`RSA15`*: RSA 15 • *`RSA OEP`*: RSA OEP
keyType	Key ID type according to *WS Security X.509 certificate token profile* **Possible values:** • *`ISSUER SERIAL`*: Issuer and serial number • *`BST DIRECT REFERENCE`*: Binary security token • *`SKI KEY IDENTIFIER`*: Subject Key Identifier • *`X509 KEY IDENTIFIER`*: X.509 issuer and serial number
signatureAlgorithm	Signature algorithm **Possible values:** • *`RSA with SHA`*: Use RSA with SHA • *`DSA with SHA1 (DSS)`*: Use DSA with SHA1 (DSS)
actor	Actuator/role used by the safety engine according to the SOAP specification.

keyAlias	Key or certificate alias in the keystore **Possible values:** Any string
keyPassword	Password for the keystore alias **Possible values:** Any string
parts	Message parts to be encrypted or signed **Possible values:** String with element name, namespace and the encryption method in quotes, separated by semicolon **Example:** `"Element";"";"Content"`
embededName	Embedded name **Possible values:** Any string

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.20.6 X4 Streaming Support

? Unknown Attachment `X4 Streaming Support`: Takes an XML or XML fragment (well-nested) and creates a document from it that can be used as output for the ReST starter. Within this XML(-fragment) `Stream` elements in the namespace `x4feature:reststream` are searched for. These are included in the output, while all other parts are Base64 encoded. Thus XML documents can be generated as ReST output in which a stream is embedded Base64-encoded (e.g. a SOAP response if MTOM is not used).

Properties

Operation	Determines which operation the adapter executes **Possible values:** `convertXML`: Convert XML

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.reststreamsupport.RestStreamSupportAdapter*: Main class (default)

inputEncoding	Character encoding of the input document **Possible values:** Any valid character encoding (e.g. *UTF-8* (default))

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

Input

The adapter input can be structured as follows:

```
<?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>
```

Output

The following XML structure is output by the adapter:

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
    <Status code="200"/>
    <Headers/>
    <Content encoding="base64">PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPF
Jvb3RFbGVtZW50PgoJPFNv
    bWVFbGVtZW50QmVmb3JlPlRleHQ8L1NvbWVFbGVtZW50QmVmb3JlPgoJPENvbnRhaW5lckZvclRo
    ZVN0cmVhbT4=<Stream xmlns="x4feature:reststream" url="xstore://SomeProject/
someData.xml" transcoding="base64-UTF-8"></Stream>
    PC9Db250YWluZXJGb3JUaGVTdHJlYW0+Cgk8U29tZUVsZW1lbnRBZnRlcj5Nb3JlIFRleHQ8L1Nv
    bWVFbGVtZW50QWZ0ZXI+CjwvUm9vdEVsZW1lbnQ+
    </Content>
</Response>
```

This output can be returned to the ReST starter immediately, or can still be provided with appropriate headers.

Content of xstore://SomeProject/someData.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Hallo><Welt/></Hallo>
```

Result of the ReST response

```
<?xml version="1.0" encoding="UTF-8"?>
<RootElement>
    <SomeElementBefore>Text</SomeElementBefore>
    <ContainerForTheStream>PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxIYW
xsbz48V2VsdC8+PC9IYWxsbz4=</ContainerForTheStream>
    <SomeElementAfter>More Text</SomeElementAfter>
</RootElement>
```

Currently there is no support when a stream is to be inserted into an attribute. In this case, the corresponding element and attribute must be assembled with Base64 strings in the `Content` element.

10.20.7 XOP (MTOM)

? Unknown Attachment `XOP (MTOM)`: Generates XOP/MTOM packages from XML documents, dissolves XOP references from an XOP/MTOM package and outputs an XML document.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *`CleanupXml`*: Removes `<?xop-candidate?>` instructions and normalizes their parent elements • *`MakeSimplepart`*: Removes `<?xop-candidate?>` instructions and normalizes their parent elements • *`MakeMultipart`*: Generate XOP/MTOM package from an XML document; The output structure is a MIME XML (see also MIME Multipart) • *`UnpackMultipart`*: Dissolve XOP references from a MIME XML, which represents an XOP/MTOM package; Output is an XML document (the actual payload)

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.xop.Adapter*: Main class (default)
RequiredMinimumPartSavings	Number of bytes that needs to be saved per element, in order for an element to be extracted via XOP/MTOM • any integer, e.g. *300* • `0` (default) • `<0`: Inefficiencies are accepted; That means, it is accepted that more bytes will be transferred, as if not extracted • `<= -300`: The extraction is performed regardless of the savings (extracting has an overhead of approx. 260–270 bytes per extraction)
RequiredMinimumTotalSavings	Specifies how many bytes need to be saved through extraction, in order for the extraction to be performed. • `0` (default) • `<0`: The result MIME document may be bigger than a result document, for which no extraction was performed. With this parameter a high negative value in `RequiredMinimumPartSavings` can be relativized.

State values

1	The operation was performed without problems
0	The operation was not executed or it was not executed completely
-1	An error occurred while processing; It is typically caused by invalid/incomplete input documents.

10.20.7.1 Operation CleanupXml

The adapter operation `CleanupXml` performs normalization works for the creation of an XOP package. Thereby, neither a candidate (identified by `<?xop-candidate?>`) will be outsourced, nor a Mime XML document will be created.

The `mimetype` of the resulting X4 document (see Document Properties Changer) is determines as follows.

Incoming *mimetype*	*Incoming encoding*	XML document is ...	Resulting *mimetype*	Resulting *encoding*
empty	*any*	no SOAP	`text/xml`	`UTF-8`
`some/type`	*any*	no SOAP	`some/type`	`UTF-8`
any	*any*	SOAP 1.1	`text/xml`	`UTF-8`
any	*any*	SOAP 1.2	`application/soap+xml`	`UTF-8`

Example

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

The adapter operation `MakeSimplepart` performs normalization works for the creation of an XOP package, whereby no candidate (identified by `<?xop-candidate?>`) will be outsourced. The result is a Mime XML document with only one part (no Multipart).

The Mime part has two headers:

- `content-transfer-encoding: binary`
- `content-type:` Wert siehe unten

Incoming *mimetype*	Incoming *encoding*	XML document is ...	Resulting `content-type` header
empty	*any*	no SOAP	`text/xml; charset=UTF-8`
`some/type`	*any*	no SOAP	`some/type; charset=UTF-8`
any	*any*	SOAP 1.1	`text/xml; charset=UTF-8`
any	*any*	SOAP 1.2	`application/soap+xml; charset=UTF-8`

Example

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

The adapter operation *MakeMultipart* is used to optimize the transfer of XML documents with bigger binary (Base64) contents. Main application areas are SOAP web services, but also any other XML document.

Input: Example of an input XML for the operation MakeMultipart

```
<SomeXml>
    <blob1>
        <?xop-candidate?>
            SGFsbG8=
    </blob1>
    <blob2>
        <?xop-candidate text/plain?>
            SGFsbG8gSGFsbG8=
    </blob2>
</SomeXml>
```

Explanation

- The XML processing instruction `<?xop-candidate?>` specifies, which text node is to be extracted
- If a media type is specified as parameter, e.g. `<?xop-candidate text/plain?>`, this media type will be used
- If no media type is specified as parameter, `application/octet-stream` will always be assumed as media type
- The adapter parameters `RequiredMinimumPartSavings` and `RequiredMinimumTotalSavings` specify how big the savings need to be in order to perform an extraction from the input XML.

Output: Example of a generated XML document with 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">
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxTb21lWG1sPgoJPGJsb2IxPjx4b3A6S
W5jbHVkZSBocmVmPSJjaWQ6MUAxOGE0ODI3ZC1lZWY3LTQ5NjUtYmY3OS1lN2YwZDBjYTI5OTQiIHhtbG5zOn
hvcD0iaHR0cDovL3d3dy53My5vcmcvMjAwNC8wOC94b3AvaW5jbHVkZSIvPjwvYmxvYjE+Cgk8YmxvYjI+PHh
vcDpJbmNsdWRlIGhyZWY9ImNpZDoyQDkyYzhlNjM5LTkwZjktNDJjNC1hMGNhLTNkOTdiODQzZTQ4YSIgeG1s
bnM6eG9wPSJodHRwOi8vd3d3LnczLm9yZy8yMDA0LzA4L3hvcC9pbmNsdWRlIi8+PC9ibG9iMj4KPC9Tb21lW
G1sPg0K</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>
```

This output XML corresponds to a MIME multipart document in which the the first Part corresponds to the input XML and all further Parts correspond to the extracted Base64 inclusions of the input XML.

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--
```

If no extractions were performed, e.g. because there were no markings or there were not enough savings, the adapter outputs the status `0` for this operation.

10.20.7.4 Operation UnpackMultipart

The adapter operation `UnpackMultipart` corresponds to the inverse operation to `MakeMultipart`. It dissolves the `xop:Include` elements from a MIME XML, which represents an XOP/MTOM package. The result is the payload XML of the XOP package (e.g. the SOAP envelope), whereby the `xop:Include` elements were replaced by the corresponding Base64 strings.

The operation also accepts a MIME XML, which is only a Part. In this case, the Part is only unpacked. However, in this case it is no guarantee that the result is an XML, e.g. if a Part corresponding to a PDF document is sent, the result will be the PDF.

In case of `xop:Include` references, only the URL protocol `cid:` is supported. The parts referenced have to be placed within the same multipart like the root part, that means they need to be sibling nodes of the root part.

Input: Example input for the 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 is an XML (or any document, if a MIME Part XML is sent)

Output: Example of an output XML

```
<SomeXml>
    <blob1>SGFsbG8=</blob1>
    <blob2>SGFsbG8gSGFsbG8=</blob2>
</SomeXml>
```

If not all `xop:Include` elements were replaced, because the references could not be dissolved, the adapter outputs the status `0` and the output is the payload XML with all replacements that could be performed.

The status `-1` is used to indicate different errors within the MIME XML.

10.20.8 YAML Converter

`YAML Converter`: Converts a YAML document into an XML document and vice versa.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Convert YAML to XML`: Convert a YAML document into an XML document • `Convert XML to YAML`: Convert an XML document into a YAML document

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** • *de.softproject.x4.adapter.yaml.YamlAdapter*: Main class (default)
`processStrings`	Controls whether numeric types and booleans are read and normalized by the adapter (*`true`*) or simply treated as literal strings (*`false`*) **Possible values:** • `true`: Numeric types and booleans are read and normalized • `false`: Numeric types and booleans are treated as literal strings

Status values

1	The conversion was successful and the output is generated
-1	An error occurred during the conversion and no output is generated

Input

Depending on the selected operations, the adapter expects the following input structure:

- Convert YAML to XML:

YAML Input

```
example:
    element1:
    - value1
    - value2
    - childValue1: 1
      childValue2: 2
    element2: value3
```

- Convert XML to YAML:

XML Input

```
<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>
```

Output

Depending on the selected operations, the adapter outputs the following structure:

- 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`: Lists the processes and resources deployed on the server in the X4 repository. The listing also includes processes deployed using X4 Apps.

Properties

`Operation`	Operation performed by the adapter **Possible values:** • `List`: Lists all available processes and resources in an XML document. • `Get Image`: Returns an image of the process workflow. • `Get Image Metadata`: Returns the metadata of an image of the process workflow. • `Reload Repository`: Returns a list of all projects and repository types for each element. • `Reload Project`: Returns the project and the type of the repository for each element.

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible value:** *de.softproject.x4.adapter.deployed.app.DeployedAppAdapter (Standard)*
`repository`	Repository name
`project`	Project name
`processPath`	Path and name of the process

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

ⓘ The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

10.21.1.1 Examples

Output List Operation

The adapter generates an XML document as 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>
```

Output Get Image Metadata Operation

```
<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>
```

Output Reload Project Operation

```
<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>
```

Output Reload Repository Operation

```
<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`: Manages the global parameters within the defined global parameter storage.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `AddKeys`: Adds all parameters to the respective parameters groups • `Export`: Exports all parameters from the parameter storage (doesn't need any input document) • `Get`: Outputs all parameters of the requested parameter groups, taking into account the cache • `Import`: All parameters within the respective parameter groups are added or replaced • `Refresh`: Reloads all parameters from the database • `RemoveKeys`: Removes all parameters within the respective parameter groups • `RemoveGroups`: Removes all parameter groups • `ReplaceGroups`: Replaces all parameters within the respective parameter groups

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.plugins.globalparameter.adapter.GlobalParameterStorageAdapter*: Main class (default)

Status values

`1 (successful)`	The operation was successful

`-1 (failed)`	The operation failed due to a technical error

ⓘ The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Input

The operations *`Get`*, *`AddKeys`*, *`RemoveKeys`*, *`RemoveGroups`*, *`ReplaceGroups`* and *`Import`* expect the following input XML documents:

Operation: Get

The following input XML structure loads the groups `DEV` and `DEFAULT` from the global parameter storage:

```
<?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEV"/>
    <Group name="DEFAULT"/>
</Placeholder>
```

Operation: AddKeys, ReplaceGroups and Import

The following XML document describes parameters for adding, replacing and importing:

```
<?xml version="1.0" encoding="UTF-8"?>
<Placeholder><!-- For operation Import optional attribute (default=false):
deleteAndImport=true|false -->
    <Group name="DEV">
        <Key name="message" type="String">Hello X4 developer!</Key>
        <!--  valid types: String|Password|Boolean|Integer|Decimal|Date -->
    </Group>
    <Group name="DEFAULT">
        <Key name="message" type="String">Hello world!</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>
```

ⓘ An empty parameter group when calling the operation `Replace` causes the entire parameter group to be emptied. Thus, the call corresponds to the operation `Remove`. For the operation `Import`, it is optionally possible to delete the corresponding parameter groups before the import. To do this, the attribute `deleteAndImport` in the root node must be set to *`true`*.

Operation: RemoveKeys

The following XML document removes single parameters of a group:

```
 <?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEFAULT">
        <Key name="endOfTheWorld"/>
    </Group>
</Placeholder>
```

Operation: RemoveGroups

The following XML document removes a complete parameter group:

```
 <?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEFAULT" />
</Placeholder>
```

Output

The operation `Refresh` outputs the current state of all parameters in an XML document.

All other operations return the parameter groups requested or changed by the request:

```
<?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEV">
        <Key name="message" type="String">Hello X4 developer!</Key>
    </Group>
    <Group name="DEFAULT">
        <Key name="message" type="String">Hello world!</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`: Manages users, groups, and roles in Keycloak.

> **Note**
> If a **Realm** other than the **Default X4 Realm** is created in **Keycloak** (http://localhost:8085/auth), the corresponding role must be assigned to the respective **Client** (`http://localhost:8085/auth/admin/master/console/#/`**`YourRealm`**`/clients`) in **Service Account Roles**.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `Activate User` • `Create new User` • `Deactivate User` • `Delete User` • `Get User by Id` • `List Users` • `Update User` • `Add Member to Group` • `Create new Group` • `Create new Sub-Group` • `Delete Group` • `Get Group by Id` • `Get Members of Group` • `List Groups` • `Remove Member from Group` • `Update Group` • `Add Composite to Role` • `Add Roles to Group` • `Add Roles to User` • `Create new Role` • `Delete Role` • `Get Composite of Role` • `Get Role by Id` • `Get Role by Name` • `Get Users with Role` • `List Roles` • `Remove Roles From a Group` • `Remove Roles from a User` • `Update Role`

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.auth.keycloak.KeycloakAdapter*: Main class (Default)
`realm`	Realm to be used in Keycloak REST API calls.
`clientId`	Client ID for Keycloak REST API authentication.

`clientSecret`	Client Secret for Keycloak REST API authentication.

Status Values

`1`	The adapter operation was executed successfully.
`-1`	An error occurred during the execution of the adapter.

The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Additional information

- Users
- Groups
- Roles

10.21.3.1 Users

The operations can be used to manage users in Keycloak.

For more information, see the section

- Create new User
- List Users
- Get User by Id
- Update User
- Activate User
- Deactivate User
- Delete User

Create new User

The `Create new User` operation creates a new user in Keycloak. The data is defined in the input XML expected by the adapter.

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	Description	Possible values
`User`		The element contains the following elements: • `Username` • `FirstName` (optional) • `LastName` (optional) • `Email` (optional) • `EmailVerified` (optional) • `Enabled` (optional) • `Attributes` (optional) • `Credentials` (optional)
`Username`	The user name desired by the user	Any string
`FirstName`	The first name of the user	Any string
`LastName`	The last name of the user	Any string
`Email`	The e-mail address of the user	E-mail address
`EmailVerified`	Specifies whether the specified e-mail address is verified.	• `true`: The email address is verified. • `false`: The email address is not verified.

Element	Description	Possible values
`Enabled`	Specifies whether the user is enabled or disabled	• `true`: The user is enabled. • `false`: The user is disabled.
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)
`Credentials`	User credentials	The element contains the following element: • `Credential`

Attributes

Element	Description	
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)
`Attribute`	Personalized, configurable attribute	The element contains the following elements: • `Name` • `Values`
`Name`	Attribute name	Any string
`Values`	Attribute values	The element contains the following elements: • `Value` (Multiple declaration possible)
`Value`	Attribute value	Any string

Credentials

The `Credentials` element defines the type of credential and the associated value. This can be a password, for example.

Element	Description	Possible values
`Credentials`	User credentials	The element contains the following elements: • `Credential`
`Credential`	Login data set	The element contains the following elements: • `Type` • `Value` • `Temporary`

Element	Description	Possible values
`Type`	Type of login data set, e.g. password	`password`: The login data set is a password.
`Value`	Attribute value	Any string
`Temporary`	Specifies whether the password is temporary	`true`: The password is temporary. `false`: The password is not temporary.

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>
```

Examples

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>
```

List Users

The `List Users` operation lists all users that exist in Keycloak.

Input structure

The adapter does not expect any input. However, without input, only the first 100 users are returned. To list more users, use the following input:

```
<KeycloakListUsersParameters>
<briefRepresentation></briefRepresentation>
<firstName></firstName>
<lastName></lastName>
<username>demo_1</username>
<email></email>
<offset></offset>
<limit>500</limit>
</KeycloakListUsersParameters>
```

The function of the parameters `briefRepresentation`, `firstName`, `lastName`, `username`, and `email` corresponds to the function in the X4 API (Swagger UI), which can be accessed via the X4 Control Center. You can leave these parameters empty in the input document.

The parameters `offset` and `limit` have the following function:

- `offset`: Number of users to be skipped from the beginning of the user list.
- `limit`: Maximum number of users to be returned. The specified number of users will be returned regardless of how many users are skipped with the `offset` parameter. Example: If `offset` is set to 50 and `limit` is set to 500 for 600 users, a list with the users 51–550 will be returned.

Examples

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>
```

Get User by Id

The `Get User by Id` operation outputs the data of the user with the specified ID.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Description	Possible values
Id	UUID of the user	UUID

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>
```

Examples

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>
```

Update User

The `Update User` operation updates the data of the user with the specified ID. The data is defined in the input XML expected by the adapter.

> ⚠ The username cannot be updated.

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	Description	Possible values
`User`	User definition	The element contains the following elements: • `Id` • `FirstName` (optional) • `LastName` (optional) • `Email` (optional) • `EmailVerified` (optional) • `Enabled` (optional) • `Totp` (optional) • `Attributes` (optional) • `Credentials` (optional) • `RequiredActions` (optional) • `NotBefore` (optional)
`Id`	Unique ID of the user	UUID
`FirstName`	The first name of the user	Any string
`LastName`	The last name of the user	Any string
`Email`	The e-mail address of the user	E-mail address

Element	Description	Possible values
`EmailVerified`	Specifies whether the specified e-mail address is verified.	• `true`: The email address is verified. • `false`: The email address is not verified.
`Enabled`	Specifies whether the user is enabled or disabled	• `true`: The user is enabled. • `false`: The user is disabled.
`Totp`	Specifies whether the password is invalid after the first login	• `true`: The user has to change the password after the first login. • `false`: The user does not need to change the password.
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)
`Credentials`	User credentials	The element contains the following element: • `Credential`
`RequiredActions`	Specifies whether the user is prompted for action when the user logs in.	`VERIFY_EMAIL`: The user has to verify his e-mail address after registration. `UPDATE_PROFILE`: The user has to update his user profile after logging in. `UPDATE_PASSWORD`: The user has to update his password after logging in.
`NotBefore`	The number of failed login attempts	Integer

Attributes

Element	Description	
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)
`Attribute`	Personalized, configurable attribute	The element contains the following elements: • `Name` • `Values`
`Name`	Attribute name	Any string

Element	Description	
`Values`	Attribute values	The element contains the following elements: • `Value` (Multiple declaration possible)
`Value`	Attribute value	Any string

Credentials

The `Credentials` element defines the type of credential and the associated value. This can be a password, for example.

Element	Description	Possible values
`Credentials`	User credentials	The element contains the following elements: • `Credential`
`Credential`	Login data set	The element contains the following elements: • `Type` • `Value` • `Temporary`
`Type`	Type of login data set, e.g. password	`password`: The login data set is a password.
`Value`	Attribute value	Any string
`Temporary`	Specifies whether the password is temporary	`true`: The password is temporary. `false`: The password is not temporary.

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>
```

Examples

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>
```

Activate User

The `Activate User` operation activates the user with the specified ID.

> ⚠ The user has to exist in Keycloak.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Description	Possible values
`Id`	UUID of the user	UUID

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>
```

Examples

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
```

Deactivate User

The `Deactivate User` operation deactivates the user with the specified ID.

> ⚠ The user has to exist in Keycloak.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Description	Possible values
`Id`	UUID of the user	UUID

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>
```

Examples

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
```

Delete User

The `Delete User` operation deletes the user with the specified ID.

> ⚠ The user has to exist in Keycloak.
> A deleted user cannot be restored.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Description	Possible values
`Id`	UUID of the user	UUID

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>
```

Examples

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
```

10.21.3.2 Groups

The operations can be used to manage groups in Keycloak.

For more information, see the section

- 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

Create new Group

The `Create new Group` operation creates a new group in Keycloak. The data is defined in the input XML expected by the adapter.

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	Description	Possible values
`Group`	Group definition	The element contains the following elements: • `Name` • `Attributes` (optional)
`Name`	Name of the group	Any string

Element	Description	Possible values
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)

Attributes

Element	Description	
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)
`Attribute`	Personalized, configurable attribute	The element contains the following elements: • `Name` • `Values`
`Name`	Attribute name	Any string
`Values`	Attribute values	The element contains the following elements: • `Value` (Multiple declaration possible)
`Value`	Attribute value	Any string

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>
```

Examples

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>
```

Create new Sub-Group

The `Create new Group` operation creates a new subgroup in Keycloak. The data is defined in the input XML expected by the adapter.

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	Description	Possible values
`ParentGroup`		The element contains the following elements: • `ParentId` • `SubGroup`
`ParentId`	UUID of the parent group	UUID
`SubGroup`	Subgroup definition	The element contains the following elements: • `Name` • `Attributes` (optional)

SubGroup

Element	Description	Possible values
`SubGroup`	Subgroup definition	The element contains the following elements: • `Name` • `Attributes` (optional)
`Name`	Name of the subgroup	Any string
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)

Attributes

Element	Description	
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)
`Attribute`	Personalized, configurable attribute	The element contains the following elements: • `Name` • `Values`
`Name`	Attribute name	Any string
`Values`	Attribute values	The element contains the following elements: • `Value` (Multiple declaration possible)
`Value`	Attribute value	Any string

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>
```

Examples

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>
```

List Groups

The `List Groups` operation lists all groups that exist in Keycloak.

Examples

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>
```

Get Group by Id

The `Get Group by Id` operation outputs the data of the group with the specified ID.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Description	Possible values
`Id`	UUID of the group	UUID

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>
```

Examples

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>
```

Update Group

The `Update Group` operation updates the data of the group with the specified ID. The data is defined in the input XML expected by the adapter.

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	Description	Possible values
`Group`	Group definition	The element contains the following elements: • `Id` • `Name` • `Attributes` (optional)
`Id`	UUID of the group	UUID
`Name`	Name of the group	Any string
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)

Attributes

Element	Description	
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)
`Attribute`	Personalized, configurable attribute	The element contains the following elements: • `Name` • `Values`

Element	Description	
`Name`	Attribute name	Any string
`Values`	Attribute values	The element contains the following elements: • `Value` (Multiple declaration possible)
`Value`	Attribute value	Any string

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>
```

Examples

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>
```

Delete Group

The `Delete User` operation deletes the user with the specified ID.

> ⚠ The group has to exist in Keycloak.
> A deleted group cannot be restored.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Description	Possible values
`Id`	UUID of the group	UUID

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>
```

Examples

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
```

Add Member to Group

The `Add Member to Group` operation adds a user to a group. The data is defined in the input XML expected by the adapter.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Ids>
    <UserId></UserId>
    <GroupId></GroupId>
</Ids>
```

Ids

Element	Description	Possible values
`Ids`		The element contains the following elements: • `UserId` • `GroupId`
`UserId`	UUID of the user	UUID
`GroupId`	UUID of the group	UUID

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>
```

Examples

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Ids>
    <UserId>f37db159-8c3c-4b0a-ae7f-5362dcf4ed1b</UserId>
    <GroupId>8445918e-f3ba-4c93-b3c4-7afd7b65fa57</GroupId>
</Ids>
```

Get Members of Group

The `Get Members of Group` operation outputs the members of the group with the specified ID.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Description	Possible values
`Id`	UUID of the group	UUID

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>
```

Examples

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>
```

Remove Member from Group

The `Remove Member from Group` operation removes a user from a group. The data is defined in the input XML expected by the adapter.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Ids>
    <UserId></UserId>
    <GroupId></GroupId>
</Ids>
```

Ids

Element	Description	Possible values
`Ids`		The element contains the following elements: • `UserId` • `GroupId`
`UserId`	UUID of the user	UUID
`GroupId`	UUID of the group	UUID

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>
```

Examples

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 Roles

The operations can be used to manage roles in Keycloak.

For more information, see the section

- 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

Create new Role

The `Create new Role` operation creates a new role in Keycloak. The data is defined in the input XML expected by the adapter.

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	Description	Possible values
`Role`	Role definition	The element contains the following elements: • `Name` • `Description` optional • `Composite` optional • `ContainerId` • `Attributes` (optional)
`Name`	Name of the role	Any string
`Description`	Description of the role	Any string
`Composite`	Specification of whether the role is a composite role	• `true`: The role is a composite role. • `false`: The role is not a composite role. (Default)
`ContainerId`	Name of the Realm	Any string
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)

Attributes

Element	Description	
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)
`Attribute`	Personalized, configurable attribute	The element contains the following elements: • `Name` • `Values`
`Name`	Attribute name	Any string
`Values`	Attribute values	The element contains the following elements: • `Value` (Multiple declaration possible)
`Value`	Attribute value	Any string

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>
```

Examples

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>
```

List Roles

The `List Roles` operation lists all roles that exist in Keycloak.

Examples

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>
```

Get Role by Id

The `Get Role by Id` operation outputs the data of the role with the specified ID.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Description	Possible values
`Id`	UUID of the role	UUID

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>
```

Examples

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>
```

Get Role by Name

The `Get Role by Name` operation outputs the data of the role with the specified name.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Name></Name>
```

Name

Element	Description	Possible values
Name	Name of the role	Any string

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>
```

Examples

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>
```

Update Role

The `Update Role` operation updates the data of the role with the specified ID. The data is defined in the input XML expected by the adapter.

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	Description	Possible values
`Role`	Role definition	The element contains the following elements: • `Id` • `Name` • `Description` (optional) • `Composite` (optional) • `ClientRole` (optional) • `ContainerId` (optional) • `Attributes` (optional)
`Id`	UUID of the role	UUID
`Name`	Name of the role	Any string
`Description`	Description of the role	Any string
`Composite`	Specification of whether the role is a composite role	• `true`: The role is a composite role. • `false`: The role is not a composite role. (Default)
`ClientRole`	Specifies whether the role is a client role. This is basically a role namespace specified for the client.	Any string
`ContainerId`	Name of the Realm	Any string
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)

Attributes

Element	Description	
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)
`Attribute`	Personalized, configurable attribute	The element contains the following elements: • `Name` • `Values`
`Name`	Attribute name	Any string
`Values`	Attribute values	The element contains the following elements: • `Value` (Multiple declaration possible)
`Value`	Attribute value	Any string

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>
```

Examples

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>
```

Delete Role

The `Delete Role` operation deletes the role with the specified ID.

> ⚠ A deleted role cannot be restored.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Description	Possible values
`Id`	UUID of the role	UUID

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>
```

Examples

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>2302cf2f-9b29-4d62-9c48-67ac5e3b0ddc</Id>
```

Add Composite to Role

The `Add Composite to Role` operation adds a child role to a composite role. Once a user is assigned the composite role, all child roles in the composite role are automatically assigned to the user.

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	Description	Possible values
`ParentId`	UUID of the composite role	UUID
`SubRoles`	Role definition of the child role	The element contains the following elements: • `SubRole` (Multiple declaration possible)

SubRoles

Element	Description	Possible values
`SubRoles`	Role definition of the child role	The element contains the following elements: • `SubRole` (Multiple declaration possible)

Element	Description	Possible values
`SubRole`	Role	The element contains the following elements: • `Id` • `Name` • `Description` (optional) • `Composite` (optional) • `ContainerId` • `Attributes` (optional)
`Id`	UUID of the role	Any string
`Name`	Name of the role	Any string
`Description`	Description of the role	Any string
`Composite`	Specification of whether the role is a composite role	• `true`: The role is a composite role. • `false`: The role is not a composite role. (Default)
`ContainerId`	Name of the Realm	Any string
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)

Attributes

Element	Description	
`Attributes`	Personalized, configurable attributes	The element contains the following elements: • `Attribute` (Multiple declaration possible)
`Attribute`	Personalized, configurable attribute	The element contains the following elements: • `Name` • `Values`
`Name`	Attribute name	Any string
`Values`	Attribute values	The element contains the following elements: • `Value` (Multiple declaration possible)
`Value`	Attribute value	Any string

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>
```

Examples

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>
```

Get Composite of Role

The `Get Composite of Role` operation outputs the child roles of a composite role with the specified ID.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

Id

Element	Description	Possible values
`Id`	UUID of the role	UUID

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>
```

Examples

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>4915229f-7544-4701-b1dc-6092861d9102</Id>
        <Name>x4_admin_access_1</Name>
        <Composite>false</Composite>
        <ClientRole>false</ClientRole>
        <ContainerId>X4Realm</ContainerId>
        <Attributes/>
    </Role>
    <Role>
        <Id>5915229f-7544-4701-b1dc-6092861d9103</Id>
        <Name>x4_admin_access_2</Name>
        <Composite>false</Composite>
        <ClientRole>false</ClientRole>
        <ContainerId>X4Realm</ContainerId>
        <Attributes/>
    </Role>
</Roles>
```

Add Roles to User

The `Add Roles to User` operation assigns one or more roles to a user.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<User>
    <UserId></UserId>
    <Roles>
        <Role>
            <Name></Name>
            <Id></Id>
        </Role>
    </Roles>
</User>
```

User

Element	Description	Possible values
`UserId`	UUID of the user	UUID
`Roles`	Role definitions	The element contains the following elements: • `Role` (Multiple declaration possible)

Roles

Element	Description	Possible values
`Roles`	Role definitions	The element contains the following elements: • `Role` (Multiple declaration possible)
`Role`	Role definition	The element contains the following elements: • `Id` • `Name`
`Id`	UUID of the role	UUID
`Name`	Name of the role	Any string

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>
```

Examples

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>
```

Get Users with Role

The `Get Users with Role` operation outputs the data of users assigned to the specified role.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Name></Name>
```

Name

Element	Description	Possible values
Name	Name of the role	Any string

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>
```

Examples

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>
```

Remove Roles from a User

The `Remove Roles from a User` operation removes assigned roles of the specified user.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<User>
    <UserId></UserId>
    <Roles>
        <Role>
            <Id></Id>
            <Name></Name>
        </Role>
    </Roles>
</User>
```

User

Element	Description	Possible values
`User`	User definition	The element contains the following elements: • `UserId` • `Roles`
`UserId`	UUID of the user	UUID
`Roles`	Role definitions	The element contains the following elements: • `Role` (Multiple declaration possible)

Roles

`Roles`	Role definitions	The element contains the following elements: • `Role` (Multiple declaration possible)
`Role`	Role definition	The element contains the following elements: • `Id` • `Name`
`Id`	UUID of the role	UUID
Element	**Description**	**Possible values**
`Name`	Name of the role	Any string

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>
```

Examples

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>
```

Add Roles to Group

The `Add Roles to Group` operation assigns one or more roles to a group.

Input XML

```
<Group>
    <GroupId></GroupId>
    <Roles>
        <Role>
            <Id></Id>
            <Name></Name>
        </Role>
    </Roles>
</Group>
```

Group

Element	Description	Possible values
`Group`	Group definition	The element contains the following elements: • `GroupId` • `Roles`
`GroupId`	UUID of the group	UUID
`Roles`	Role definitions	The element contains the following elements: • `Role` (Multiple declaration possible)

Roles

Element	Description	Possible values
`Roles`	Role definitions	The element contains the following elements: • `Role` (Multiple declaration possible)
`Role`	Role definition	The element contains the following elements: • `Id` • `Name`
`Id`	UUID of the role	UUID
`Name`	Name of the role	Any string

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>
```

Examples

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>
```

Remove Roles From a Group

The `Remove Roles from a Group` operation removes assigned roles from the specified group.

Input XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Group>
    <GroupId></GroupId>
    <Roles>
        <Role>
            <Id></Id>
            <Name></Name>
        </Role>
    </Roles>
</Group>
```

Group

Element	Description	Possible values
`Group`	Group definition	The element contains the following elements: • `GroupId` • `Roles`
`GroupId`	UUID of the group	UUID
`Roles`	Role definitions	The element contains the following elements: • `Role` (Multiple declaration possible)

Roles

`Roles`	Role definitions	The element contains the following elements: • `Role` (Multiple declaration possible)
`Role`	Role definition	The element contains the following elements: • `Id` • `Name`
`Id`	UUID of the role	UUID
Element	**Description**	**Possible values**
`Name`	Name of the role	Any string

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>
```

Examples

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`: Lists all process steps and execution times of a process and returns detailed information about a specific process.

> **Note:**
> The output of the adapter contains the information that is present in the profiling at the time the output is generated. Depending on the configuration, the profiling data is deleted after a certain period of time. If longer periods are to be taken into account and persistence is desired, this must be configured accordingly. For more information, refer to the X4 Control Center Guide > Process Monitor > Profiling.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `List Process Executions`: Return processes and execution times of a process • `Find Process Execution Detail`: Return detailed information about a process with the specified UUID • `List Process Execution Steps Detail`: Return process steps of a process with the specified UUID

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.plugins.profiler.ProcessProfilerAdapter*: Main class (Default)

Status values:

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

> Documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Additional information:

- Operation: Find Process Execution Detail
- Operation: List Process Executions
- Operation: List Process Execution Steps Detail

10.21.4.1 Operation: Find Process Execution Detail

The `Find Process Execution Detail` operation outputs detailed information about a process with the specified PID.

Examples

Input

The adapter expects an XML document as input:

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>15c28be0-d02b-4ca9-be08-b4096afef7f6</Id>
```

The following elements have to be included in the input:

Element	Description
Id	UUID of the process

Output

The adapter generates an XML document as 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>
```

For more information, see the section

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.4.2 Operation: List Process Executions

The operation `List Process Executions` lists process and execution times of a process.

⚠ A filter has to be specified as input.

The adapter expects the following structure as input XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<Query>
  <Paging limit="" offset=""/>
  <!-- FILTER -->
</Query>
```

The following elements are included in the input:

Element	Description
`Paging`	Paging is used to output data only in subsets.
`Query`	Filter

✓ For more information, see the section

- Filter by all subprocesses of a process
- Filter by process path
- Filter by process status
- Filter by subprocesses of a process
- Filter by time period
- Sorting data in output
- Output data in subsets

Filter by all subprocesses of a process

To filter by all subprocesses of a process including the subprocesses of the subprocesses, the following elements can be declared:

Element	Description
`ProcessFilterByRootId`	Filter type **Contains the following elements:** • `ProcessStatus` (optional) • `RootProcessInstanceId`

Element	Description
`ProcessStatus`	Process status, multiple declaration possible. **Possible values:** • `RUNNING` • `PAUSED` • `FINISHED` • `ERROR` • `TERMINATED` • `CRASHED`
`RootProcessInstanceId`	UUID of the process

Examples

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>
```

Filter by process path

To filter by the process path, the following elements must be declared:

Element	Description
`ProcessFilterByPath`	Filter type **Contains the following elements:** • `ProzessStatus` (optional) • `ProcessPath`

Element	Description
`ProcessStatus`	Process status, multiple declaration possible. **Possible values:** • `RUNNING` • `PAUSED` • `FINISHED` • `ERROR` • `TERMINATED` • `CRASHED`
`ProcessPath`	Path to the X4 process to be executed **Possible values:** Any valid path to the X4 process in the X4 repository (e.g. `1/MyProject/Processes/MyProcess_1.wrf`)

Examples

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>
```

Filter by process status

To filter by process status, the following elements must be declared:

Element	Description
`ProcessFilter`	Filter type **Contains the following elements:** • `ProcessStatus`

Element	Description
`ProcessStatus`	Process status, multiple declaration possible. **Possible values:** • `RUNNING` • `PAUSED` • `FINISHED` • `ERROR` • `TERMINATED` • `CRASHED`

Examples

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>
```

Filter by subprocesses of a process

To filter for subprocesses of a process, the following elements can be declared:

Element	Description
`ProcessFilterByParentProcess`	Filter type **Contains the following elements:** • `ProcessStatus` (optional) • `CallingProcessReference`

Element	Description
`ProcessStatus`	Process status, multiple declaration possible. **Possible values:** • `RUNNING` • `PAUSED` • `FINISHED` • `ERROR` • `TERMINATED` • `CRASHED`
`CallingProcessReference`	Filter definition **Contains the following elements:** • `ProcessInstanceId` • `ActionId`
`ProcessInstanceId`	UUID of the process
`ActionId`	ID of the process step ⓘ The `ActionId` can be taken from the output of the `Find Process Execution Detail` operation.

Examples

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>
```

Filter by time period

To filter by a time period, the following elements can be declared:

Element	Description
`Period`	Time period specification (according to ISO 8601) **Required attributes:** • `from`: Start date of the time period • `to`: End date of the time period

Example

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>
```

Sorting data in output

You can sort the data in the output according to certain criteria, either in ascending or descending order.

You enable sorting by declaring the `<Sorting>` element in the input XML. Within the `<Sorting>` element, the sorting criteria are defined in the `<Field>` element.

Element	Description
`Sorting`	Sorting **Contains the following elements:** • `Field`
`Field`	**Contains the following elements:** • `Name` • `Order`
`Name`	Sorting criterion **Possible values:** • `StartedAt`: Start date • `FinishedAt`: End date • `ProcessInstanceId`: ID of the process • `ProcessPath`: Process path • `ParentProcessInstanceId`: ID of the parent process • `ParentActionId`: ID of the parent process step • `RootProcessId`: ID of the root process • `HostName`: Host name • `TotalTime`:Total duration • `Status`: Status

Element	Description
`Order`	Sorting order **Possible values:** • `ASC`: Ascending • `DESC`: Descending

Examples

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>
```

Output data in subsets

You can output the data in subsets, e.g. to minimize the processing load.

To do this, declare the `<Paging>` element in the input XML. You configure the subset with the attributes of the element.

Attribute	Description
`offset`	Defines the offset of the data query, i.e. from which data set the data source is to be read. **Possible values:** Integer
`limit`	Number of data sets to be included in the response. **Possible values:** Integer greater than `0`

Examples

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

The operation `List Process Execution Steps Detail` outputs process steps of a process with the specified UUID.

Examples

Input

The adapter expects the following structure as input XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<StepQuery>
  <ProcessInstanceId></ProcessInstanceId>
  <StepFilter>
    <TechnicalName></TechnicalName>
    <DisplayName></DisplayName>
    <Status></Status>
  </StepFilter>
  <Paging limit="" offset=""/>
</StepQuery>
```

The following elements are included in the input:

StepQuery

Element	Description
`ProcessInstanceId`	UUID of the process
`StepFilter`	Filter (optional)
`Paging`	Paging is used to output data only in subsets. (Optional)

StepFilter

Element	Description
`TechnicalName`	Technical name of the process module
`DisplayName`	Label of the process module
`Status`	Status value of the process module

```
<?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 to change the runtime configuration for the monitoring of X4 processes.

> A monitoring defines for which processes within a project information is logged. Thus, for example, certain processes can be excluded from logging.
> Only one `.profiling` file can be created in a project.
> The information about executed and logged processes can be read out with the Process Monitoring adapter.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `List Profilings`: List all profilings • `Get Profiling by Id`: Return profiling with the specified UUID. • `Update Profiling`: Update profiling with the specified UUID.

Properties

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.core.profiling.runtime.ProfilingsAdapter*: Main class (Default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Additional information:

- Operation List Profilings
- Operation Get Profiling by Id
- Operation Update Profiling

10.21.5.1 Operation List Profilings

The `List Profilings` operation lists all profilings within the X4 repository.

Examples

Output

The adapter generates an XML document as 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>
```

The following elements can be included in the output:

Element	Description
Id	UUID of the project in which a profiling `.profiling` is located.
RepositoryName	Name of the X4 repository folder
ProjectName	Name of the project in the X4 repository

For more information, see the section

- Operation List Profilings
- Operation Get Profiling by Id
- Operation Update Profiling

10.21.5.2 Operation Get Profiling by Id

The `Get Profiling by Id` operation outputs the profiling that is in the project with the specified Id.

Examples

Input

The adapter expects an XML document as input:

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>f30c0ce7-fbc5-4988-81d4-7de74cb4dca8</Id>
```

The following elements have to be included in the input:

Element	Description
Id	UUID of the project where the profling `.profiling` is located.

Output

The adapter generates an XML document as 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>
```

The following elements can be included in the output:

Element	Description
Id	UUID of the project where the profling `.profiling` is located.
Processes	List of processes for which profiling rules exist.
@filterType	Attribute: Defines whether the processes listed below are explicitly included or excluded from logging. Possible values: • `include` • `exclude`
Process	Element with relative path to the related process for which profiling rules apply. Multiple declaration possible.

For more information, see the section

- Operation List Profilings
- Operation Get Profiling by Id
- Operation Update Profiling

10.21.5.3 Operation Update Profiling

The `Update Profiling` operation updates the Profiling located in the project with the specified Id.

The changes are updated only at runtime. The file of the file type `.profiling` in the repository is not updated.

Examples

Input

The adapter expects an XML document as 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>
```

The following elements have to be included in the input:

Element	Attribute	Description
`Profiling`	`active`	**Possible values:** • `true`: Activates process profiling for the current project. • `false`: Disables process profiling, identical with the behavior without the `.profiling` file within the corresponding project.
`Id`		UUID of the project where the profiling `.profiling` is located.
`Processes`		Parent element of single `Process` elements listed in it. (optional) ⓘ If no Processes is specified, the profiling with the filterType `include` refers to all processes of the current project.
`Processes`	`filterType`	**Possible values:** • `include` The following process list should be logged explicitly, all others are not logged. • `exclude` The following process list should be explicitly excluded from logging, all others will be logged. ⓘ Only one `Processes` element with a corresponding `filterType` can be specified.
`Process`		Single process for which profiling rules apply. Multiple declaration possible.
`Process`	`path`	Relative path within the corresponding repository project. **Possible values:** String

For more information, see the section

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.6 Process Runtime Management

`Process Runtime Management`: Stops, terminates and lists running processes.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `List Running Processes`: List all running processes • `Stop Process`: Stop process with specified UUID, the current process step is executed • `Terminate Process`: Try to terminate the process with specified UUID immediately, the current process step is aborted

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.plugins.management.ProcessRuntimeAdapter*: Main class (Default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Additional information

- Operation: List Running Processes
- Operation: Stop Process
- Operation: Terminate Process

10.21.6.1 Operation: List Running Processes

The `List Running Processes` operation lists all processes running on the X4 server.

Examples

Output

The adapter generates an XML document as 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>
```

The following elements can be included in the output:

Element	Description
Process:id	UUID of the running process
Repository	Name of the X4 repository folder
Project	Name of the folder where the process is located in the X4 repository.
Process	Name of the running process
StartedAt	Start date and start time of the current process

For more information, see the section

- Operation: List Running Processes
- Operation: Stop Process
- Operation: Terminate Process

10.21.6.2 Operation: Stop Process

The `Stop Process` operation stops the running process with the specified PID. The current process step is executed.

Examples

Input

The adapter expects an XML document as input.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<id>f744ffa1-960a-4122-a407-3dcdbd2a7924</id>
```

The following elements have to be included in the input:

Element	Description
`Id`	UUID of the running process

For more information, see the section

- Operation: List Running Processes
- Operation: Stop Process
- Operation: Terminate Process

10.21.6.3 Operation: Terminate Process

The `Terminate Process` operation attempts to terminate the running process with the specified UUID immediately. The current process step is terminated.

Examples

Input

The adapter expects an XML document as input.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<id>f744ffa1-960a-4122-a407-3dcdbd2a7924</id>
```

The following elements have to be included in the input:

Element	Description
`Id`	UUID of the running process

For more information, see the section

- Operation: List Running Processes
- Operation: Stop Process
- Operation: Terminate Process

10.21.7 Project Parameter Management

`Project Parameter Management`: This adapter contains an API for reading and changing project parameters at runtime so that they can be adapted to the system environment.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `Get Project Parameter IDs`: Returns the IDs of the projects containing parameters with workspaces and project names • `Get Project Parameter Structure`: Returns the XML structure of the project parameters with groups • `List Project Parameters`: Returns the XML structure of the project parameters • `Patch Project Parameters`: Updates one or more project parameters for the specified project ID

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.core.parameter.runtime.ProjectParametrizationAdapter*: Main class (Default)
`withSystemPrivileges`	**Possible values:** Boolean (true/false) False: Restricts the access to read and edit parameters depending on access rights (roles). True: No restriction to read and edit parameters depending on access rights (roles). This setting must be used if a process is called outside of a user's context.

Status values

`1 (successful)`	The operation was executed successfully.
`0 (empty)`	The operation was performed successfully, but no parameter was found or changed.
`-1 (failed)`	The operation failed due to a technical error.

The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Additional information:

- 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

The `Get Project Parameter Ids` operation returns the IDs of the projects containing parameters with workspaces and project names.

Input

The adapter does not expect any input during this operation.

Example

Output

The adapter generates an XML document as 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>
```

The following elements can be included in the output:

Element	Attribute	Description
`ProjectParameter`		Project parameter with ID, name and workspace
	`Id`	UUID of the project containing a `.param` file with project parameters
	`Workspace`	Name of the workspace in which the project is stored

Element	Attribute	Description
	`ProjectName`	Name of the project in the X4 repository

The data displayed by the operation can be further utilized in a mapping (e.g. reading out the ID of the current project).

For more information, see the section

- 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

The `Get Project Parameter Structure` operation expects a `UUID` as input and returns the XML structure of the project parameters with project parameter groups.

Example

Input

The adapter expects an XML document as input:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Id>efb66be5-9ff3-4c1a-9879-9f46854694c0</Id>
```

Output

The adapter generates an XML document as 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>
```

The following elements can be included in the output:

Element	Attribute	Description
`Parameters`		Parameters of a project with ID, name and workspace of the project
	`Id`	UUID of the project in which a **Project Parameter Configuration** file `.param` is contained.
	`ProjectName`	Name of the project in the X4 repository
	`Workspace`	Name of the workspace in which the project is stored.
`Enums`		
`Group name`		Name of the project parameter group
`Variable parameter`		Project parameter (variable)
`Constant parameter`		Projekt parameters (constant)
	`completeName`	Name of the parameter with parameter group
	`name`	Name of the parameter
	`type`	Data type of project parameter
`Description`		Description of the project parameter
`Value`		Default value of the project parameter

For more information, see the section

- 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

The operation `List Project Parameters` expects a `UUID` as input and returns the XML structure of the parameters for the project with the given `UUID`.

Example

Input

The adapter expects an XML document as input:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Id>efb66be5-9ff3-4c1a-9879-9f46854694c0</Id>
```

Output

The adapter generates an XML document as 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>
```

The following elements can be included in the output:

Element	Attribute	Description
`Parameters`		Project parameter with ID, name and workspace of the project
	`Id`	UUID of the project containing a `.param` file with project parameters
	`ProjectName`	Name of the project in the X4 repository
	`Workspace`	Name of the workspace in which the project is stored

Element	Attribute	Description
`Variable`		Type of project parameter
`Constant`		Type of project parameter
	`completeName`	Name of parameter with parameter group

For more information, see the section

- 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

The operation `Patch Project Parameters` updates one or more project parameters for the specified project ID.

The changes are only updated at runtime. The file of file type `.param` in the repository is not updated.

If the **Patch Project Parameters** operation fails, no parameters are updated.

Updating variables and constant
Project parameters of type **Constant** cannot be updated.
If project parameters of type **Variable** and **Constant** are included in the input XML file, no parameters are updated.

Example

Input

The adapter expects an XML document as 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>
```

The following elements must be included in the input:

Element	Attribute	Description
`Parameters`		Parameter of a project, with ID, project name and workspace
	`Id`	UUID of the project containing a `.param` file with project parameters
	`ProjectName`	Name of the project in the X4 repository
	`Workspace`	Name of the workspace in which the project is stored
`Variable`		Type of project parameter
`Constant`		Type of project parameter
	`completeName`	Name of parameter with parameter group

Output

The operation generates an output with the changed parameters.

For more information, see the section

- 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`: This adapter contains an API to access and work with savepoints.

Properties

`Operation`	Using this dropdown list, you specify the operation to be executed by the adapter. **Possible values:** • `List Savepoints`: list all savepoints • `Resume Savepoint`: resume a process after a stop triggered by a savepoint • `Delete Savepoint`: delete savepoint

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values:** *de.softproject.x4.plugins.management.SavePointAdapter*: main class (default)

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Additional information:

- Operation: List Savepoints
- Operation: Delete Savepoint
- Operation: Resume Savepoint

10.21.8.1 Operation: List Savepoints

The `List Savepoints` operation generates a list of the savepoints that have been set in a process.

Input structure

The adapter does not expect any input.

Example

Output

The adapter generates an XML file as 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>
```

The following elements can be included in the output:

Element	Description
`Savepoint`	The UUID of the savepoint is specified with the `id` attribute.

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td>PID</td><td>Process ID

Note:

There are two scenarios for getting the PID:

Scenario 1
Savepoints always get the PID of the root process when the process containing a savepoint is called via the root process. This scenario is shown in the above output. Here, the PIDs are identical for all the savepoints retrieved.

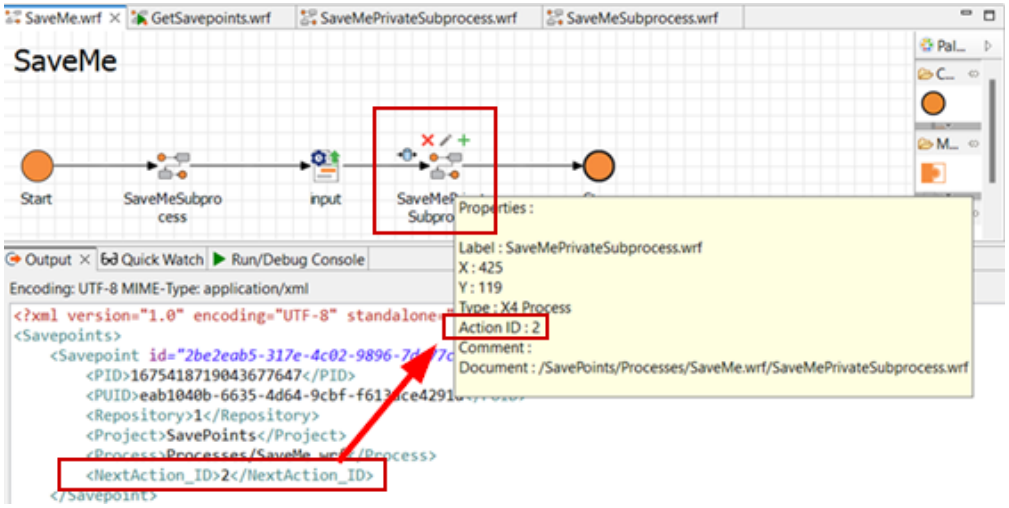
Scenario 2
If the process with the savepoint is not called with the root process, the PID of the process with the savepoint is used.

Example for scenario 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 of the process which was used to get the savepoints

Note:

For savepoints created with an X4 version prior to 7.3, only the PID is retrieved, not the PUID.</td></tr>
<tr><td>Repository</td><td>The workspace where the project is stored</td></tr>
<tr><td>Project</td><td>Name of the project which contains the Technical Process</td></tr>
<tr><td>Process</td><td>Name of the Technical Process where the savepoints are set</td></tr>
</table>

Element	Description
`NextAction_ID`	The Action ID of the element following the savepoint that was set in the process specified in the output Example for the first savepoint from the above output: 

Additional information:

- Operation: List Savepoints
- Operation: Delete Savepoint
- Operation: Resume Savepoint

10.21.8.2 Operation: Delete Savepoint

The `Delete Savepoint` operation deletes a set savepoint.

Input XML structure

The adapter expects an XML document as input.

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

The following element must be included in the input:

Element	Description	Possible values
`Id`	UUID of the savepoint	UUID

Example

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>cced145e-9056-4eb2-bbcf-756a92215c78</Id>
```

Output

The operation does not generate a specific output, i.e., the output is the same as the input.

Additional information:

- Operation: List Savepoints
- Operation: Delete Savepoint
- Operation: Resume Savepoint

10.21.8.3 Operation: Resume Savepoint

The `Resume Savepoint` resumes a process after a stop triggered by a savepoint.

Input XML structure

The adapter expects an XML document as input.

```
<?xml version="1.0" encoding="UTF-8"?>
<Id></Id>
```

The following element must be included in the input:

Element	Description	Possible values
`Id`	UUID of the savepoint	UUID

Example

Input

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>65ece1ae-182c-4c5b-9116-dc1f4c03355e</Id>
```

Output

The operation does not generate a specific output, i.e., the output is the same as the input.

Additional information:

- Operation: List Savepoints
- Operation: Delete Savepoint
- Operation: Resume Savepoint

10.21.9 Schedule Management

`Schedule Management`: Updates or lists schedulers that are in the X4 repository or in a project.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `List Schedules`: List scheduler with all schedules • `Get Schedule by Id`: Output scheduler with the specified Id • `Update Schedule`: Update scheduler with the specified Id • `Jobs Information`: Displays information about scheduled processes from all Schedule files in the X4 repository.

Parameter

`Adapter`	Main adapter class (do not change!) **Mögliche Werte:** *de.softproject.x4.plugins.scheduler.runtime.SchedulesAdapter*: Main class (Default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Additional information

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.9.1 Operation: List Schedules

The `List Schedules` operation lists all schedules within the X4 repository.

Examples

Output

The adapter generates an XML document as 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>
```

The following elements can be included in the output:

Element	Description
Id	UUID of the project in which the scheduler `.schedule` is located.
RepositoryName	Name of the X4 Repository folder
ProjectName	Name of the project in the X4 repository

For further information please visit the section

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.9.2 Operation: Get Schedule by Id

The `Get Schedule by Id` operation outputs the scheduler located in the project with the defined Id.

Examples

Input

The adapter expects an XML document as input:

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>f30c0ce7-fbc5-4988-81d4-7de74cb4dca8</Id>
```

The following elements must be included in the input:

Element	Description
Id	UUID of the project in which the scheduler `.schedule` is located.

Output

The adapter generates an XML document as 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>
```

The following elements can be included in the output:

Element	Description
Id	UUID of the project in which the scheduler `.schedule` is located.
Cron	Execution type `Cron`
Startup	Execution type `Startup`
Startup:Id	UUID of the execution
Cron:Id	UUID of the execution
Name	Name of the execution
Process	Path to the process to be executed
Active	Execution status **Possible values:** *`true`* / *`false`*
CronExpression	Cron expression of the execution, only relevant for the execution type `Cron`
StartDate	Start date of the execution, only relevant for the execution type `Startup`
EndDate	End date of the execution, only relevant for the execution type `Startup`

For further information please visit the section

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.9.3 Operation: Update Schedule

The `Update Schedule` operation updates the scheduler located in the project with the defined Id.

The changes are updated only at runtime. The `.schedule` file type in the repository is not updated.

Examples

Input

The adapter expects an XML document as 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>
```

The following elements must be included in the input:

Element	Description
Id	UUID of the project in which the scheduler `.schedule` is located.
Cron	Execution type `Cron`
Startup	Execution type `Startup`
Startup:Id	UUID of the execution
Cron:Id	UUID of the execution
Active	Execution status **Possible values:** *`true`* / *`false`*
CronExpression	Cron expression of the execution, only relevant for the execution type `Cron`
StartDate	Start date of the execution, only relevant for the execution type `Startup`
EndDate	End date of the execution, only relevant for the execution type `Startup`

For further information please visit the section

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.9.4 Operation: Jobs Information

The Operation `Jobs Information` displays information about scheduled processes from all Schedule files in the X4 repository.

Examples

Output

The adapter generates an XML document as output.

Note

The format for date and time depends on the date and time format in the system.

Possible formats:

- DD.MM.YY, hh:mm
- MM/DD/YY, hh:mm AM/PM

Output in the 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, 12:30</StartTime>
        <PreviousTime>26.04.22, 12:30</PreviousTime>
        <NextTime>27.04.22, 12:30</NextTime>
    </Job>
</JobInformation>
```

Output in the 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>
```

The following elements can be included in the output:

Element	Description	Possible values
`Id`	UUID of the execution	• UUID
`Name`	Name of the execution	• Any string
`Process`	Path to the process to be executed	• Any valid path to the X4 process in the X4 repository

Element	Description	Possible values
`Active`	Execution status	• `true` • `false`
`JobsExecuted`	Number of previous executions	• Integer
`StartTime`	Execution start date	• Date and time
`PreviousTime`	Date of last execution	• Date and time
`NextTime`	Date of the next execution	• Date and time
`EndTime`	Execution end date	• Date and time

For more information, see section

- Operation: List Schedules
- Operation: Get Schedule by Id
- Operation: Update Schedule
- Operation: Jobs Information

10.21.10 System Configuration Management

`System Configuration Management`: Shows the configuration parameters of the X4 instances.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `List Configurations`: List all configurations of the system • `Get Configuration by configurationName`: Output configuration with the specified name

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.plugins.management.configuration.SystemConfigurationAdapter*: Main class (Default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Additional information

- Operation: List Configurations
- Operation: Get Configuration by configurationName

10.21.10.1 Operation: List Configurations

The `List Configurations` operation lists all configurations of the system.

Examples

Output

The adapter generates an XML document as 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

The `Get Configuration by configurationName` operation outputs the configuration with the specified name.

Examples

Input

The adapter expects an XML document as input:

```
<?xml version="1.0" encoding="UTF-8"?>
<ConfigurationName>
    <Value>KeyCloakConfiguration</Value>
</ConfigurationName>
```

Output

The adapter generates an XML document as 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`: Lists the license information of third-party libraries.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `Get Summary`: List license information of third-party libraries

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.plugins.management.license.ThirdPartyLicenseAdapter*: Main class (Default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Additional information

- Operation: Get Summary

10.21.11.1 Operation: Get Summary

The `Get Summary` operation lists the license information of third-party libraries.

Examples

Output

The adapter generates an XML document as 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`: Updates or lists Configuration files that are in the X4 repository or in a project.

For more information about Configuration in Web Apps, see the **Web App Guide**.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `List WebApp Configurations`: Output a list of all Configurations in the repository. • `Get Configuration by Id`: Output Configuration with the specified Id. • `Update Configuration`: Update Configuration with the specified Id.

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.server.webapp.adapter.WebAppConfigurationAdapter*: Main class (Default)

Status values

1	The operation was executed successfully.
-1	The operation failed due to a technical error.

The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Additional information

- Operation: Get Configuration by Id
- Operation: List WebApp Configurations
- Operation: Update Configuration

10.21.12.1 Operation: Get Configuration by Id

The `Get Configuration by Id` operation returns the configuration that is in the project with the specified Id.

If changes were made at runtime, the configuration content used at runtime is output.

Example

Input

The adapter expects an XML document as input:

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>0cbff222-12e8-43f3-a2b8-31388c50afee</Id>
```

The following elements have to be included in the input:

Element	Description
Id	UUID of the project where the configuration `.wac` is located.

Output

The adapter generates an XML document as 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>
```

The `<Configuration>` element can contain the following elements:

Possible elements:

Element	Description
`General`	Contains configuration definitions about the general appearance of the Web App.
`Access`	Contains configuration definitions for accessing the Web App.
`Status`	Contains configuration definitions for the internal status values of the Web App.
`Logout`	Contains configuration definitions for logout.

For more information, see the section

- Operation: Get Configuration by Id
- Operation: List WebApp Configurations
- Operation: Update Configuration

10.21.12.2 Operation: List WebApp Configurations

The `List WebApp Configurations` operation lists all configurations in the repository.

Example

Output

The adapter generates an XML document as 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>
```

The following elements are included in the output:

Element	Description
Id	UUID of the project where the Configuration `.wac` is located.
RepositoryName	Name of the X4 repository folder
ProjectName	Name of the project in the X4 repository

For more information, see the section

- Operation: Get Configuration by Id
- Operation: List WebApp Configurations
- Operation: Update Configuration

10.21.12.3 Operation: Update Configuration

The `Update Configuration` operation updates the configuration located in the project with the specified ID.

Note:
The changes are updated only at runtime. The `.wac` configuration file in the **Projects** view is not updated.
These attributes are not changed at runtime:

- `<Notifications />`
- `<HeaderIncludes />`

Examples

Input

The adapter expects an XML document (example below) as 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>
```

The following elements have to be included in the input:

Element	Description
`Update`	The root element of the XML document.
`Id`	UUID of the project where the Configuration `.wac` is located.
`Configuration`	The root element of the `.wac` file
`SecurityHTTPHeaders`	Used to protect the Web App against unwanted attacks.

The attributes of the elements can be found in the Web App Guide, in the section Declaring Configuration.

Read more:

- Operation: Get Configuration by Id
- Operation: List WebApp Configurations
- Operation: Update Configuration

10.21.13 Web App Theme Management

`Web App Theme Management`: Updates or lists Web App Theme files located in the X4 repository or in a project.

For more information on Theming (Fonts and Colors), refer to the **Web App Guide**.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `List WebApp Themes`: Output list with all Web App Theme in repository • `Get Theme by Id`: Output Web App Theme with the specified Id • `Update Theme`: Update Web App Theme with the specified Id

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.server.webapp.adapter.WebAppThemeConfigurationAdapter*: Main class (Default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Additional information

- Operation: Get Theme by Id
- Operation: List WebApp Themes
- Operation: Update Theme

10.21.13.1 Operation: Update Theme

The `Update Theme` operation updates the theme located in the project with the specified Id.

The changes are updated only at runtime. The theme file of `.theme` file type in the repository is not updated.

For more information on Theming (Fonts and Colors), refer to the **Web App Guide**.

Examples

Input

The adapter expects an XML document as 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>
```

The following elements have to be included in the input:

Element	Description
`Update`	The root element of the XML document.
`Id`	UUID of the project where the theme file of `.theme` file type is located.
`Theme`	The root element of the theme file.

Status values

`1`	The adapter operation was executed successfully.
`-1`	An error occurred while executing the adapter.

Additional information

- Operation: Update Theme
- Operation: Get Theme by Id
- Operation: List WebApp Themes

10.21.13.2 Operation: Get Theme by Id

The `Get Theme by Id` operation outputs the Theme file located in the project with the specified Id.

If changes were made at runtime, the contents of the theme file used at runtime are output.

For more information on Theming (Fonts and Colors), refer to the **Web App Guide**.

Examples

Input

The adapter expects an XML document as input:

```
<?xml version="1.0" encoding="UTF-8"?>
<Id>fa2c23b6-1843-4758-b686-06f3f748afff</Id>
```

The following elements have to be included in the input:

Element	Description
`Id`	UUID of the project that contains the theme file with the `.theme` file type.

Output

The adapter generates an XML document as 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>
```

The <Theme>element can contain the following elements:

Possible elements

Element	Description
`ColorScheme`	Contains definitions of the Main Theme used in the Web App.
`ChartScheme`	Contains definitions of the chart theme used in the Web App.
`Fonts`	Contains definitions of the fonts used in the Web App.

Additional information

- Operation: Update Theme
- Operation: Get Theme by Id
- Operation: List WebApp Themes

10.21.13.3 Operation: List WebApp Themes

The `List WebApp Themes` operation lists all projects with a theme file in the repository.

For more information on Theming (Fonts and Colors), refer to the **Web App Guide**.

Examples

Output

The adapter generates an XML document as 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>
```

The following elements are included in the output:

Element	Description
`Id`	UUID of the project where the configuration `.wac` is located.
`RepositoryName`	Name of the X4 repository folder.
`ProjectName`	Name of the project in the X4 repository.

Additional information

- Operation: Update Theme
- Operation: Get Theme by Id
- Operation: List WebApp Themes

10.22 X4 Tools

The X4 tools category bundles all adapters that provide functionality for use in technical processes.

- 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`: Interface to the Case Management functionality of the product component X4 BPM. The adapter enables you to manage documents for processes. In addition to characteristics of the business data model, documents can be stored directly in processes.

Properties

`Operation`	Defines the operation executed by the adapter. **Possible values:** • `AttachDocument`: Defines the operation executed by the adapter. • `UpdateDocument`: Updates an existing document. • `UpdateDocumentMetadata`: Updates the metadata of an existing document. • `GetDocument`: Returns the document as output. • `GetDocumentMetadata`: Returns the metadata of the document as output. • `ListAllDocumentMetadata`: Returns the metadata of all documents of the case as output. • `RemoveDocument`: Removes the document from the case.

Parameters

`Adapter`	Main class of the adapter (Do not change!) **Possible values:** *de.softproject.x4.server.bmn.casemanagement.adapter.CaseDocumentAdapter*: Main class (default)
`caseId`	ID of the case to be processed **Possible values:** • Any (valid) Case ID • `%CASE_ID%`: current Case ID (default)
`title`	Title of the document **Possible values:** Any string
`fileName`	File name of the document **Possible values:** Any string, e.g. `Appendix/Invoice.pdf`
`encoding`	Character encoding of the document **Possible values:** Valid specification of character encoding, e.g. `UTF-8`
`mimeType`	MIME Type of the document **Possible values:** Valid specification of the MIME Types, e.g. `application/xml`

`base64`	Defines if the document is passed as a Base64 string. **Possible values:** • `true`: Document is passed as Base64 string. • `false`: Document is passed directly (default).

Status values

`1`	The operation was executed successfully.
`-1`	An error occurred during the operation's execution.

Input

The adapter doesn't expect any specific input.

> If the parameter `base64` is set to *`true`* in the `AttachDocument` and `UpdateDocument` operations, the input must be a Base64 encoded string.

Output:

The adapter returns different kinds of documents depending on the executed operation:

- In the following operations, the input is output again if the operation is successful:
  - `AttachDocument`
  - `UpdateDocument`
  - `UpdateDocumentMetadata`
  - `RemoveDocument`
- `GetDocumentMetadata` operation

  Sample Output for the GetDocumentMetadata Operation

```
<?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>Invoice</Title>
    <FileName>Invoice.xml</FileName>
    <MimeType>application/xml</MimeType>
    <Encoding>UTF-8</Encoding>
</Document>
```

- `ListAllDocumentMetadata` operation

Sample Output for the ListAllDocumentMetadata Operation

```
<?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>Comment.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>Invoice</Title>
        <FileName>Invoice.xml</FileName>
        <MimeType>application/xml</MimeType>
        <Encoding>UTF-8</Encoding>
    </Document>
</Documents>
```

- `GetDocument` operation:
  If the parameter `base64` is set to *`true`*, the document will be output as Base64 encoded string.

10.22.2 Case Management

`Case Management`: Forms the interface to the "Case Management" functionality of the product component X4 BPM. The adapter enables, for example, the setting and reading of case data such as case status or characteristics of the case data model.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `SetFeature`: Writes a feature • `SetFeatures`: Writes several features in one operation • `GetFeatures`: Returns all features of the operation • `SetState`: Sets the status of the operation • `GetStates`: Returns all possible states of the operation • `GetComments`: Returns all comments of the operation • `GetInformation`: Returns all information of the operation

Parameter

`Adapter`	Main class of the adapter (do not change!) **Possible values**: *en.softproject.x4.server.bmn.casemanagement.adapter.CaseManagementAdapter*: Main class (default)

`CaseId`	ID of the operation to be processed **Possible values:** • Any valid user ID • *`%CASE_ID%`*: Current case ID (default)
`CaseProcess`	Repository path to a Business Process in which the corresponding operation is defined. **Possible values:** • Any complete repository path, e.g. *`Project/Folder/Process.bpm`* • Any valid placeholder, e.g. *`%START_PROCESS%`* • *`%START_PROCESS%`*: Calling Business Process (default)
`CaseUserId`	User ID of the owner of an operation **Possible values:** • Any valid user ID • *`%USER_ID%`*: Current user ID (default)
`dataformatVersion`	Version of the data format used **Possible values:** • `1.0`: Data format used up to version 7.0 • `2.0`: Used data format from version 7.0, see input examples

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed but no result was found or no data was changed.
`-1`	The operation could not be executed.

Input

The adapter expects a predefined XML structure as input. The structure depends on the operation used.

If the attribute "type" is not specified for setFeature / setFeatures, the data type is "String". The other possible data types of the "Case Management" functionality correspond to those of the types there.

- `SetFeature`-Operation:

Expected structure for the operation SetFeature

```
<?xml version="1.0" encoding="UTF-8" ?>
<Name>Max</Name>
```

- `SetFeatures`-Operation:

Expected structure for the 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:

Expected structure for the operation SetState

```
<?xml version="1.0" encoding="UTF-8" ?>
<State>NEW</State>
```

Output:

Depending on the operation, the adapter outputs data in different XML structures:

- `GetFeatures`-Operation

Example output for the operation GetFeatures

```
<?xml version="1.0" encoding="UTF-8" ?>
<Features>
    <Name>Max</Name>
    <Lastname>Mustermann</Lastname>
    <Address>Musterweg 1</Address>
</Features>
```

- `GetStates`-Operation

Example output for the operation GetStates

```
<?xml version="1.0" encoding="UTF-8" ?>
<CaseStates>
    <State>NEW</State>
    <State>DONE</State>
</CaseStates>
```

- `GetInformation` operation

Example output for the 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`: Provides statistics for cases (Business Process) and human tasks.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `GetAllCaseMetadata`: Provide all cases metadata which corresponds to the specified filter criterion • `GetAllTaskMetadata`: Provide all task metadata which corresponds to the specified filter criterion

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.server.bpm.casemanagement.adapter.CaseManagementStatisticsAdapter* Main class (default)

Status values

1	The operation was executed successfully.
-1	The operation failed due to a technical error.

Input

The adapter expects a specific XML document as input depending on the selected operation.

Operation GetAllCaseMetadata

```
<?xml version="1.0" encoding="UTF-8"?>
<CaseMetadataFilter>
    <!-- Filters all cases for the specified start date -->
    <From>2019-10-01T08:00:00.000</From>
    <!-- Filters all cases for the specified end date. -->
    <To>2019-10-07T08:00:00.000</To>
    <!-- Filters all cases for the specified process name -->
    <ProcessName>Invoice/Processes/InvoiceApproval.bpm</ProcessName>
    <!-- Filters all cases for the specified case state -->
    <CaseState>Approved</CaseState>
    <!-- Filters all cases for the specified technical state
        possible values: RUNNING, CRASHED, WAITING, FINISHED, FAILED, UNKNOWN -->
    <TechnicalState>FINISHED</TechnicalState>
</CaseMetadataFilter>
```

Operation GetAllTaskMetadata

```
<?xml version="1.0" encoding="UTF-8"?>
<TaskMetadataFilter>
    <!-- Filters all human tasks for the specified start date. -->
    <From>2019-10-01T08:00:00.000</From>
    <!-- Filters all human tasks for the specified end date. -->
    <To>2019-10-07T08:00:00.000</To>
    <!-- Filters all cases for the specified process name -->
    <ProcessName>Invoice/Processes/InvoiceApproval.bpm</ProcessName>
    <!-- Filters all human tasks for the specified task state
    possible values: OPEN, CLOSED, IN_PROGRESS, DEADLINE_REACHED -->
    <TaskState>OPEN</TaskState>
    <!-- Filters all human tasks for the specified case state -->
    <CaseState>To be approved by clerk</CaseState>
    <!-- Filters all human tasks for the specified deadline -->
    <Deadline>2019-10-07T08:00:00.000</Deadline>
    <!-- Filters all human tasks for the specified assigned user or group (technical
identifier must be specified) -->
    <Assignment type="GROUP">clerks</Assignment>
    <!-- Filters all human tasks for the specified owner -->
    <Owner>emil</Owner>
</TaskMetadataFilter>
```

⚠ Note that when filtering for groups, i.e. `Assignment` of the type *GROUP*, the technical identifier and not the group name must be specified.

Output

The adapter outputs a specific XML document depending on the selected operation.

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>Upload protocol</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>Release invoice</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`: Calls an external application or an external command via a Command Line of the operating system with any number of arguments. The adapter expects an adapter-specific input XML structure with Base64-encoded data and outputs an XML document with the Base64-encoded results of `StdOut` and/or `StdErr`, as well as the returned exit code.
Former name: *OS Process Execution Adapter*

Properties

Operation	Defines the operation executed by the adapter **Possible values:** *`Execute`*: Call application/command via Command Line

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.osprocess.OsProcessAdapter*: Main class (default)
WorkingDirectory	Working directory for the executed application call; might be overwritten by the working directory specified within the input XML; If both are empty, the *X4 Server* working directory will be used. **Possible values:** • Any valid path (e.g. *C:\X4\temp*) • *(empty)*: No default working directory (set manually via input XML, else the X4 Server working directory will be used. (default)
CollectStdOut	Capture the Standard Output stream of the called program **Possible values:** • *true*: Save Base64-encoded Standard Output stream of the called program within the result XML document in element <StdOut> (default) • *false*: Do not consider the Standard Output (element <StdOut> in the result XML document remains empty)
CollectStdErr	Capture Standard Error stream of the called program **Possible values:** • *true*: Save Base64-encoded Standard Error stream of the called program within the result XML document in element <StdErr> (default) • *false*: Do not consider the Standard Error (element <StdErr> in the result XML document remains empty)
MergeStdErrIntoStdOut	Output both Base64-encoded Standard Error stream and Base64-encoded Standard Output stream together within the element <StdOut> **Possible values:** • *true*: Output Standard Error and Standard Output merged together • *false*: Output data streams separately (default)

Input

The adapter expects an XML document having the following pattern:

```
<Command>
    <WorkingDirectory>
        <!--Working directory (optional)-->
    </WorkingDirectory>
    <Application>
        <!-- Command line name or complete path of the application -->
    </Application>
    <!-- Any number of arguments (observe sequence!) -->
    <Argument>
        <!--Argument/Parameter-->
    </Argument>
    <Environment clear="Work only with the following environment variables (true/
false)">
        <!-- any number of environment variables -->
        <Entry name="Environment variable name"><!--Wert--></Entry>
    </Environment>
    <Input>
        <!-- Base64-encoded data to be processed by the external application -->
    </Input>
    <Copy>
        <!-- Any string/XML structure to be bypassed without modification -->
    </Copy>
</Command>
```

Explanation:

- `<WorkingDirectory>` (optional): Working directory in which the called application operates; overwrites the value of parameter `WorkingDirectory`); does not correspond to the path of the application (see below)!
- `<Application>`: Application or command line name with or without the complete path (if the application path is already set within the environment variables). As an alternative, use `cmd.exe` under Windows and specify the application as first argument.
- `<Environment>`: Defines environment variables; Using the attribute `clear="true"` you can control, whether the command will be executed only with the environment variables defined within the element `<Environment>`. With `clear="false"` the environment variables defined within the element `<Environment>` will be added to the existing ones.
- `<Input>` (optional) Contains Base64-encoded input data
- `<Copy>` (optional): Contains an XML structure or any string to be bypassed without modification.

Unlike in case of adapter parameters, you can use placeholders like `%USER_COLLECTION%` wit hin the input XML!

Example

Within the following input XML structure, the Windows default Command Line `cmd.exe` will be executed in the working directory `C:\`, and the environment variables `name` and `age` are set using the

command argument `set`. With the argument `/c` the command will be executed in character order and then it will be terminated. With `clear="true"` the command will be executed only with the environment variables defined here.

```
<Command>
  <WorkingDirectory>C:\</WorkingDirectory>
  <Application>cmd.exe</Application>
  <Argument>/c</Argument>
  <Argument>set</Argument>
  <Environment clear="true">
    <Entry name="name">Theo Test</Entry>
    <Entry name="age">35</Entry>
  </Environment>
  <Input></Input>
  <Copy>
  </Copy>
</Command>
```

Output

The function adapter outputs an XML document with the following pattern:

```
<CommandResult>
   <ExitCode>Exit code of the application</ExitCode>
   <StdOut>Base64-encoded output data of the application</StdOut>
   <StdErr>Base64-encoded error output of the application</StdErr>
   <Copy>String/XML structure bypassed without modification</Copy>
</CommandResult>
```

Status values

1	Call has been executed; the application's exit code equals 0.
0	Call has been executed; the application's exit code does not equal 0.
-1	Call could not be executed (e. g. application was not found).

10.22.5 Document Comparison

`Document Comparison`: Compares a repository file with an input file and outputs a status value as process result.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** *Compare*: Compare files with each other

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.comparison.X4ComparisonAdapter*: Main class (default)
comparisonMethod	Comparison method **Possible values:** • `BYTE`: Compare files byte by byte (default) • `ZIP`: Compare ZIP files by their content • `XML`: Compare XML files by their content • `TEXT`: Compare text files by their content • `EMPTY`: Compare input files with empty arrays of bytes
expectedResult	Path to the file to be compared with the adapter input; Relates to the comparison methods `BYTE`, `ZIP`, `XML` and `TEXT` **Possible values:** Repository path, e. g. *`X4Experience/BPM/1-Examples/InvoiceReceipt/BPM/Data/Invoice.xml`* (mandatory field)
ignoreWhitespace	Determines whether whitespaces will be considered while comparing file contents; Relates to the comparison methods `XML` and `TEXT` **Possible values:** • `true`: Ignore whitespaces • `false`: Consider whitespaces (default)
normalizeWhitespace	Determines whether whitespaces in `TEXT` or `XML` document will be normalized; Relates to the comparison methods `XML` and `TEXT` **Possible values:** • `true`: Normalize whitespaces (default) • `false`: Do not normalize whitespaces
ignoreComments	Determines whether comments within the XML document are considered while comparing file contents; Relates to the comparison method `XML` **Possible values:** • `true`: Ignore comments • `false`: Consider comments (default)

compareSimilar	Determines whether the order of tags within the XML document is considered while comparing file contents, e. g. if the XML document's content is the same but in another order; Relates to the comparison method `XML` **Possible values:** • `true`: Ignore order • `false`: Consider order (default)
ignoreCDataAndTextDiff	Determines whether the content within the `CDATA` element will be interpreted as plain text content; Relates to the comparison method `XML` **Possible values:** • `true`: Interpret the content of `CDATA` as plain text (default) • `false`: Do not interpret the content of `CDATA` as plain text

Input

The adapter expects as input any file, which is compared with a specified Repository file.

Output

The adapter outputs one of the following status values as process result:

`1`	The compared documents are identical
`0`	The compared documents are different
`-1`	The documents could not be compared with each other, since the input was either empty, not well formed or false.

10.22.6 Document Properties Changer

`Document Properties Changer`: Reads and changes document properties of a so-called `X4Document` used in the X4 ESB for process data exchange. Only the internally used document properties are changed, which may influence the execution of some adapters (but not XSL mappings). The actual data of the document remains unchanged.

Former name: *Document Properties Adapter*

> To change the character encoding of the document data, use the Charset Changer . To convert characters of the document into another character encoding, use the Encoding Converter .
> Also note: Changes in the character encoding made with the *Document Properties Changer* do not affect XSL transformations. If the document does not contain any XML declaration with a character encoding, *UTF-8* will be used as default character encoding according to the XML specification.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** *`Change`*: Change MIME type and/or character encoding of the `X4Document`

Parameters

`Adapter`	Main adapter class (Do not change!) **Possible values:** *`de.softproject.integration.adapter.documentproperties.DocumentPropertiesAdapter`*: Main class (default)
`mimeType`	New MIME type of the X4Document **Possible values:** Any MIME type (e.g. *`text/xml`*)
`encoding`	New character encoding of the `X4Document` **Possible values:** Any character encoding (e.g. *`ISO-8859-1`*)
`removeEncoding`	Removes the character encoding of the `X4Document` ⚠ If the `removeEncoding` parameter is enabled, the `encoding` parameter must be empty.

Status values

`1`	The adapter operation was executed successfully.
`-1`	An error occurred during the execution of the adapter.

10.22.7 Empty Output Adapter

`Empty Output Adapter`: Empties the data stream in the Technical Process and thus serves as a data sink to create a well-defined state within the Technical Process. Especially in production mode of the *X4 server*, this adapter is useful to create an empty file without requiring access to a file system. This processes any input data and outputs a 0 byte binary document or nothing (Null).

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *`Empty`*: Output document with 0 byte length • *`Null`*: Output nothing (Null)

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.sink.Sink*: Main class (default)

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

10.22.8 Global Context Manipulation

`Global Context Manipulation`: Stores similar to context variables any document within a Technical Process temporarily into a global hash table in order to provide this context also for other processes. The documents stored in the global context will be saved under a key, and are available at the runtime of the *X4 Server*. The function adapter is able to create, overwrite and delete entries in the hash table, and to list all or only selected keys from the hash table.

> A context is valid throughout the whole server, and all Technical Processes access the same hash table. Thus, it is possible for other threads to put in a document under the same key at the same time. In addition, small time differences between the operations `Put` and `PutGet` can output ambiguous (e.g. empty) results.
> Within XSL mappings, you can access already existing context keys using the `globalcontext:` URL protocol at runtime.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • *Get*: Read entry whose key is specified within the parameter Key fr om the global context and output the last completely saved document • *Put*: Store the input document in the global context under the key specified within the parameter Key, and output this entry • *PutGet*: Store the input document in the global context under a key, and output the last completely saved document (it can't be guaranteed that the most recent document will be output!) • *Remove*: Delete the entry whose key is specified within the parameter Key from the global context and output the deleted entry • *List*: Output an XML document containing all keys from the global context • *RemoveAllMatching*: Delete all entries whose keys match a Regex filter expression specified within the parameter Filter from the global context • *KeepAllMatching*: Keep all entries from the global context whose keys match a Regex filter expression specified within the parameter Filter (and delete all other entries) • *ListAllMatchingKeys*: Output an XML document containing all keys from the global context that match a Regex filter expression specified within the parameter Filter • *ListAllNotMatchingKeys*: Output an XML document containing only keys from the global context that do not match a Regex filter expression specified within the parameter Filter

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.globalcontext.GlobalContextAdapter*: Main class (default)
Key	Key under which the entry is stored in the global context **Possible values:** Any string
Filter	Regular expression (Regex) to filter keys in the global context, according to *java.util.Pattern*, see http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html **Possible values:** Any regular expression (e.g. *.*Test.**)

Status values

1	The operation has been executed successfully.
0	The operation has been executed successfully, but the result is Null or an empty list.
-1	The adapter could not be executed.

10.22.9 Keycloak Token

`Keycloak Token`: Adapter which returns a Keycloak token so that the user can log-in a with the Keycloak user and password.

Properties

`Operation`	Operations executed by the adapter **Possible values** • `Get Token` - Request token

Parameter

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.x4.auth.keycloak.KeycloakTokenAdapter*: Hauptklasse (Standard)
`userName`	User name (mandatory)
`password`	User password (mandatory)
`realm`	Realm to be used in Keycloak REST API calls.
`clientId`	Client ID for Keycloak REST API authentication.
`clientSecret`	Client Secret for Keycloak REST API authentication.

Status Values

`1`	The adapter operation was executed successfully.
`-1`	An error occurred during the execution of the adapter.

> **Note**
> The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

10.22.10 Log4J Logging

`Log4J Logging`: Outputs data from the process via *Log4J* to the server log. All data that can be converted into a string can serve as input data. In `AdvancedMode`, data can be output directly at various log levels in an XML document with an adapter-specific structure.
This adapter transmits its input data unchanged to the next process step.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Execute`*: Execute adapter

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.log4j.Log4jAdapter`*: Main class (default)
logger	Used logger to which the log message is to be sent **Possible values:** • Any logger name • *`X4Logger`*: Output as X*`X4Logger`* in the server console via Log4j
level	Level of the output log message. In the Log4j configuration a threshold (parameter `Threshold`) is defined, from which level log messages are output; mostly this is *`INFO`*. **Possible values:** • *`DEBUG`*: Output debug information (if allowed by the logging configuration) • *`INFO`*: Output general information or processed process data • *`WARN`*: Output a warning for an unexpected situation • *`ERROR`*: Output error message (caught exception) • *`FATAL`*: Output critical system error ⓘ Log messages with the level `FATAL` are output with the level `ERROR` for technical reasons. The output receives `FATAL:` as prefix.

advancedMode	Activate extended mode in which an input XML document with an adapter-specific structure is processed separately. Such an XML document can contain log messages that are output at a specified log level (see example below) **Possible values:** • *yes*: parse input XML documents • *no*: Do not parse input XML documents (default)
modifyMessage	Modify/expand log message **Possible values:** • *Nothing*: Output log message with unchanged layout • *AddPid*: Additionally output the current process instance ID in the log message (default).

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Example

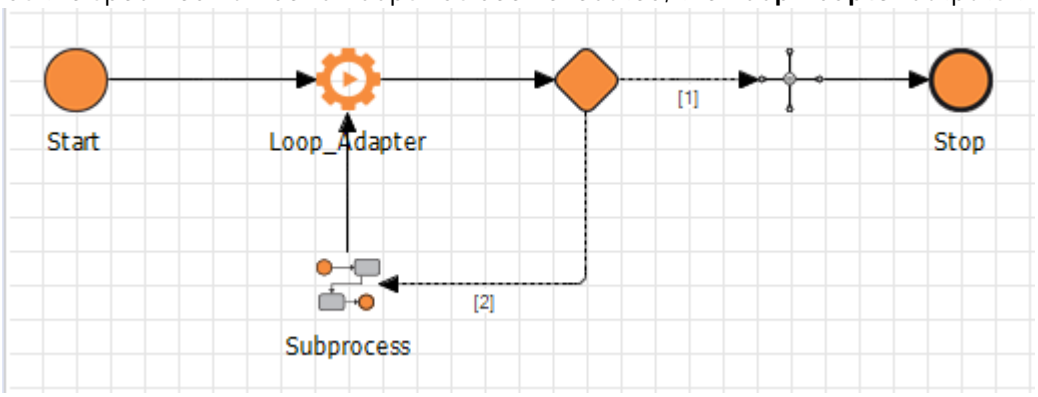
If the `advancedMode` parameter is enabled: This adapter-specific input XML document contains information about the log level and the log messages to be output. The messages are output directly to the server log:

```
<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`: The **Loop Adapter** executes a specified number of loops in a process.

As soon as the specified number of loops has been executed, the **Loop Adapter** outputs the status value `1`.



Properties

`Operation`	Operation executed by the adapter **Possible values:** • `Count`: Counts the number of loops passed.

Parameter

`count`	Number of loops to be executed. **Possible values:** • Integer

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

10.22.12 Parallel Process Starter

`Parallel Process Starter`: Executes an *X4* process in any user folder in `<X4>/X4DB` any number of times in parallel and allows to synchronize the results again afterwards. The adapter expects an input XML document with data to be processed, each stored within arbitrarily named child elements below the root element. Depending on the configuration, the adapter outputs an XML document or a Base64-encoded string.

> The *Parallel Process Starter* allows you to set the process name (and path) at runtime via *dynamic parameters*. In addition, each *X4* process started via the *Parallel Process Starter* has its own process instance ID, is executed in a separate thread and therefore cannot access variables of the calling process.

General properties and parameters

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Execute`*: Execute process(es)

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.parallelprocessstarter.Adapter*: Main class (default)
X4ProcessURL	Path to the X4 process to be executed **Possible values:** Any valid `x4db:` URL to the X4 process in the X4 Repository (e.g. *`x4db:/1/Project/Folder/Process.wrf`*)
MaxThread	Number of threads to be used **Possible values:** • Any positive integer (e.g. *3*) • *0* or negative integer: do not restrict parallel execution (e.g. *-1*)
InputIsBase64	Specifies the data format for the adapter input **Possible values:** • `True`: The content of each element is a Base64 encoded string which is decoded and passed as input to the child process. • `False`: The XML element content is passed directly to the child process.
expandOutput	Controls the output format of the adapter **Possible values:** • `True`: The result is output completely as an XML document. • `False`: The result is output as a Base64 encoded string.

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

Input

In the input XML document, an arbitrarily named root element can contain any number of child elements. For each child element, an X4 process is called and the element node is passed as input.

Sample input

```
<SomeRootElement>
  <SomeInputElement/>
  <SomeInputElement>with some content</SomeInputElement>
  <SomeInputElement someAttribute="someValue">more content</SomeInputElement>
</SomeRootElement>
```

With the parameter *InputIsBase64:False* the above example is executed as follows:

Process call 1

```
<SomeInputElement/>
```

Process call 2

```
<SomeInputElement>with some content</SomeInputElement>
```

Process call 3

```
<SomeInputElement someAttribute="someValue">more content</SomeInputElement>
```

Output

The result of the process execution could be structured as follows in the above example:

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>
```

In case of an error, no `Result` element is generated, but e.g. the following result is output:

Output in case of error

```
<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`: Executes a Technical Process in any user folder in `<X4>/X4DB` with a Technical Process instance ID synchronously or asynchronously. As input, the started process receives the input of the adapter. If the function adapter is executed *asynchronously*, then the adapter immediately outputs an XML document containing the Technical Process instance ID of the started Technical Process; if the function adapter is executed *synchronously*, then it waits for and outputs the result of the started Technical Processs (if any).

With the *Process Starter*, you can use *dynamic parameters* to set the process name (and path) at runtime. In addition, each Technical Process started via the *Process Starter Adapter* has its own Technical Process instance ID, is executed in a separate thread, and therefore cannot access variables of the calling process.

General properties and parameters

Properties

Operation	Determines which operation the adapter executes **Possible values:** *Run*: Start Technical Process

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.processstarter.ProcessStarter*: Main class (default)
User	User ID (X4DB user folder where the Technical Process to be started is located) **Possible values:** • Any valid user ID (e.g. *999*) • *1*: Default user folder (default)

Process	Repository path to a Technical Process that has the `In service: true` property. **Possible values:** Any path in the *X4* repository without leading slash (e.g. `Project/Folder/Process.wrf`)
synchron	Execute Technical Process synchronously or asynchronously **Possible values:** • `True`: Wait until the started Technical Process instance is terminated and output its result if necessary, see below (default). • `False`: Output the Technical Process instance ID in an XML document and immediately execute the following Technical Process step, see below.
forceNewThread	Start a new thread even if executed synchronously **Possible values:** • `True`: Start a new thread on synchronous execution; This option allows to better manage the state associated with the threads, e.g. for transactions. (default) • `False`: Do not start a new thread when executed synchronously (original behavior).

Synchronously started Technical Processes

If you start Technical Process synchronously (with parameter `synchron: True`), then the adapter waits until the started Technical Process has been completely executed and, if necessary, outputs its result including status code before executing the next process step.

Status values

`1`	(Or any other positive value) The adapter was successfully executed *synchronously*. The status code of the synchronously started Technical Process is taken over by the *Process Starter* or, if necessary, output modified (see note below). If the adapter outputs status `1`, it does not necessarily mean that the started Technical Process outputs a result.
`0`	The Technical Process to be started is either not active (property `In service: false`) and was therefore not executed, or empty blocks are used in the started Technical Process, or the started Technical Process outputs the status `0`.

`-1`	The adapter could not be executed (the Technical Process to be started was not found or the started Technical Process throws errors).
(Any negative value)	The started Technical Process may have been executed and generated a negative state value, e.g. via an XSL mapping with `ConditionState`, see Dynamic state values with `ConditionState`. In case of asynchronously started Technical Processes, their status is taken over by the *Process Starter*. Further process execution is aborted if the process property `Stop on error: true`.

Notice:

- If a synchronously started Technical Process returns a status value between *950* and *999*, this status is changed by the *Process Starter* (status value minus 50) and status values between *900* and *949* are output. This is necessary in order not to influence the *Save Point* function of the *X4 server*.
- If a synchronously started Technical Process returns a status value between *-999* and *-950*, this status is changed by the *Process Starter* (status value plus 50) and status values between *-900* and *-949* are output. This is necessary to avoid errors in the process execution.

Asynchronously started Technical Process

If you start Technical Processes asynchronously (parameter `synchron: False`), then after the called Technical Process is started, its process instance ID is output in an XML document and the immediately following process step is executed.

Status values

1 (successful)	The function adapter was successfully executed *asynchronously*. If the started Technical Process is not active (property In service: *false*) or throws errors, then status *1* is still output; the result XML document contains the process instance ID *-1* in these cases.
-1 (error)	The function adapter could not be executed (the Technical Process to be started was not found).

10.22.14 Projects

`Projects`: Allows access to files within a project, in process libraries and in adapters.

This adapter replaces the Repository Explorer (deprecated).

Properties

`Operation`	Defines the operation executed by the adapter. **Possible values:** • `Create`: Creates a new project. **Required parameters:** `projectName`, `projectType` • `List`: Lists all existing projects. • `Delete`: Deletes a project. **Required parameters:** `projectName`

Parameter

`Adapter`	Main class of the adapter (Do not change!) **Possible values:** *de.softproject.integration.adapter.system.repository.ProjectAdapter*: Main class (default)
`projectName`	Name of the project to be created or deleted. **Possible values:** String without spaces and special characters
`projectType`	Type of the project to be created. **Possible values:** • *`ESB Project`* • *`Web App Project`* • *`Adapter Project`* • *`Web Portal Project`*

Status values

`1`	The operation was executed successfully.
`-1`	An error occurred during the operation's execution.

Input

The adapter does not expect any input. Existing input is output unchanged.

Output

- The adapter only returns an output in the `Create` and `Delete` operations if an error occurs.
- `List` operation:

Sample Output for the List Operation

```
<Projects>
    <Project type="ESB" name="MyESBProject" />
    <Project type="BPM" name="MyBPMProject" />
</Projects>
```

10.22.15 Project Access

`Project Access`: Allows access to files within a project, in process libraries and adapters.

> This adapter replaces the Repository Explorer (deprecated).

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `ReadFile`: Read file defined in the `path` parameter. • `WriteFile`: Write data from the previous process step to the target file defined in the `path` parameter. • `AppendFile`: Append file from the previous process step to the target file defined in the `path` parameter. > If the target file defined in the `path` parameter does not exist, a file will be created. • `Exist`: Check whether the files or folders defined in the `path` parameter exist. Outputs an XML with metadata as a result. • `CreateDirectory`: Create folder defined in the `path` parameter. • `Delete`: Delete files or folders defined in the `path` parameter. • `ListDirectory`: List names and metadata of files and folders from the folder defined in the `path` parameter. Outputs an XML with metadata as a result. > The content of the files is not listed. • `Move`: Move files or folders defined in the `path` parameter to the folder defined in the `targetPath` parameter. • `Copy`: Copy files or folders defined in the `path` parameter to the folder defined in the `targetPath` parameter.

Parameters

`Adapter`	Main adapter class (Do not change!) **Possible values:** *de.softproject.integration.adapter.system.repository.ProjectAccessAdapter*: Main class (default)
`path`	Path to the file or folder to be processed. **Possible values:** Folder path ⓘ The project path must be defined without the project name. **Example**: *Processes/MyFolder*
`targetPath`	For `copy` and `move` operations: target of the file/folder to be copied or moved.
`nameFilter`	Filters the output of the `ListDirectory` operation by name **Possible values:** • Any string (e.g. *`myFolder`*) • Wildcard `*`

Status values

`1`	The adapter operation was executed successfully.
`0`	No file was found for the `Exists` operation.
`-1`	An error occurred during the execution of the adapter.

For further information please visit the section

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.1 Operation: ReadFile

The `ReadFile` operation reads the data from the file defined in the `path` parameter.

Examples

The following example shows a sample parameterization of the adapter for the `ReadFile` operation to read the data from the destination file defined in the `path` parameter.

Parameters	Value
`path`	Resources/File.xml

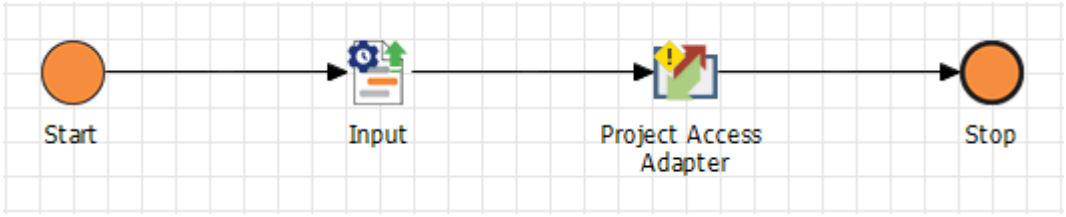
For further information please visit the section

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.2 Operation: WriteFile

The `WriteFile` operation writes data from the previous process step to the destination file in the X4 repository defined in `path` parameter.

Examples



The following example shows a sample parameterization of the adapter for the `WriteFile` operation to write the data from the previous process step to the destination file defined in `path` parameter.

Parameters	Value
`path`	Resources/File.xml

For further information please visit the section

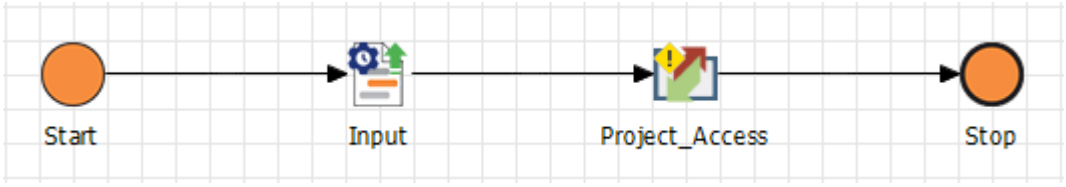
- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.3 Operation: AppendFile

The `Append File` operation appends the data from the previous process step to the destination file in the X4 Repository defined in `path` parameter.

If the target file defined in the `path` parameter does not exist, a file will be created.

Examples



The following example shows a sample parameterization of the adapter for the `Append File` operation to append the data from the previous process step to the destination file defined in `path` parameter.

Parameters	Value
path	Resources/File.xml

Target file

```
<?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>
```

⚠ The output is not a valid XML file. The output must be transformed into a valid XML file using a mapping.

For further information please visit the section

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.4 Operation: Exists

The `Exists` operation checks whether the files or folders defined in the `path` parameter exist. Outputs an XML with metadata as a result.

Status values

`1`	The file or folder exists.
`0`	The file or folder does not exist.

Examples

The following example shows a sample parameterization of the adapter for the `Exist` operation to check whether the target file defined in `path` parameter exists.

Parameters	Value
`path`	Resources/File.xml

For further information please visit the section

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.5 Operation: CreateDirectory

The `CreateDirectory` operation creates the folder defined in the `path` parameter.

Examples

The following example shows a sample parameterization of the adapter for the `CreateDirectory` operation to create the folder defined in the `path` parameter.

Parameters	Value
`path`	TemporaryFiles/MyFolder

For further information please visit the section

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.6 Operation: Delete

The `Delete` operation deletes the local file or folder defined in the `path` parameter.

The data will be permanently deleted.

Examples

Delete file

The following example shows a sample parameterization of the adapter for the `Delete` operation to delete the target file defined in the `path` parameter.

Parameters	Value
`path`	Resources/File.xml

Delete folder

The following example shows a sample parameterization of the adapter for the `Delete` operation to delete the destination folder defined in the `path` parameter.

Parameters	Value
`path`	Resources/MyFolder

For further information please visit the section

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.7 Operation: ListDirectory

The `ListDirectory` operation lists names and metadata of files and folders from the folder defined in the `path` parameter. Outputs an XML with metadata as a result.

`1`	The adapter operation was executed successfully.
`-1`	The folder doesn't exist.

Examples

The following example shows a sample parameterization of the adapter for the `ListDirectory` operation to list PDF files from the folder defined in the `path` parameter.

Parameters	Value
`path`	Resources/MyFolder
`nameFilter`	*.pdf

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>
```

When a folder is listed, the attribute `isDirectory="true"` is set.

For further information please visit the section

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.8 Operation: Move

The `Move` operation moves files or folders defined in the `path` parameter to the folder defined in the `targetPath` parameter.

Examples

Move folder

The following example shows a sample parameterization of the adapter for the `Move` operation to move the `MyFolder` folder to the `X4` folder:

Parameters	Value
path	Resources/MyFolder
targetPath	Resources/X4/MyFolder

Move file

The following example shows a sample parameterization of the adapter for the Move operation to move the MyFile.xml to the MyFolder folder:

Parameters	Value
path	Resources/MyFile.xml
targetPath	Resources/MyFolder/MyFile.xml

For further information please visit the section

- Operation: ReadFile
- Operation: WriteFile
- Operation: AppendFile
- Operation: Exists
- Operation: CreateDirectory
- Operation: Delete
- Operation: ListDirectory
- Operation: Move
- Operation: Copy

10.22.15.9 Operation: Copy

The Copy operation copies the file or folder defined in the path parameter to the local folder defined in the targetPath parameter.

Examples

Copy folder

The following example shows a sample parameterization of the adapter for the Copy operation to copy the MyFolder folder to the X4 directory:

Parameters	Value
path	Resources/MyFolder
targetPath	Resources/X4/MyFolder

Copy file

The following example shows a sample parameterization of the adapter for the `Copy` operation to copy the `MyFile.xml` file to the `MyFolder` directory:

Parameters	Value
`path`	Resources/MyFile.xml
`targetPath`	Resources/MyFolder/MyFile.xml

For further information please visit the section

- 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`: The **System Status** Adapter outputs status information about the X4 Server.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `Threads Information`: Outputs information about all threads running on the X4 Server • `JVM Parameters`: Outputs all parameters of the Java Virtual Machine • `Processes Running`: Outputs information about all processes running on the X4 Server • `User Sessions Memory`: Outputs information about the memory consumption of the user sessions • `Apps Status`: Outputs information about the failed and deployed X4 Apps

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.

-1	The operation failed due to a technical error.

The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

For more information, see the section

- Operation: Threads Information
- Operation: JVM Parameters
- Operation: Processes Running
- Operation: User Sessions Memory
- Operation: Apps Status

10.22.16.1 Operation: Threads Information

The `Threads Information` operation outputs information about all threads running on the X4 Server.

Examples

Output

The adapter generates an XML document as 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>
```

The following elements are included in the output:

Element	Description	Possible values
`Name`	Name of the thread	• Any string

Element	Description	Possible values
`Id`	ID of the thread	• Integer
`State`	Thread status	• `NEW` • `RUNNABLE` • `BLOCKED` • `WAITING` • `TIMED_WAITING` • `TERMINATED`
`Priority`	Priority of the threads	• Integer betwenn `1` and `10`
`Daemon`	Specification whether the thread is a background service	• `true` • `false`
`CpuTime`	Time needed for the execution of the current thread in seconds	• Integer

For more information, see the section

- Operation: Threads Information
- Operation: JVM Parameters
- Operation: Processes Running
- Operation: User Sessions Memory
- Operation: Apps Status

10.22.16.2 Operation: JVM Parameters

The `JVM Parameters` operation outputs all parameters of the Java Virtual Machine.

Example

Output

The adapter generates an XML document as 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>
```

The following elements are included in the output:

Element	Description	Possible values
`MaxMemory`	Maximum available memory (in megabytes)	any integer

Element	Description	Possible values
`CurrentMemory`	Total memory, i.e. located memory, including used and free memory (in megabytes)	any integer
`FreeMemory`	Located unused memory (in megabytes)	any integer
`AvailableProcessors`	Number of available processors	any integer

Additional information:

- Operation: Threads Information
- Operation: JVM Parameters
- Operation: Processes Running
- Operation: User Sessions Memory
- Operation: Apps Status

10.22.16.3 Operation: Processes Running

The `Processes Running` operation outputs information about all processes running on the X4 Server.

Example

Output

The adapter generates an XML document as 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>
```

The following elements are included in the output:

Element	Description	Possible values
`UUID`	UUID of the process	UUID
`Name`	Name of the process	Any string
`Path`	Path of the process	Any string
`Module`	Name of the module that executes the process	`engine`
`MemoryConsumed`	Total memory consumed by the processes in bytes	`Double`

Additional information:

- Operation: Threads Information
- Operation: JVM Parameters
- Operation: Processes Running
- Operation: User Sessions Memory
- Operation: Apps Status

10.22.16.4 Operation: User Sessions Memory

The `User Sessions Memory` operation outputs information about the memory consumption of the user sessions.

Example

Output

The adapter generates an XML document as 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>
```

The following elements are included in the output:

Element	Description	Possible values
`SessionId`	ID of the user session	UUID

Element	Description	Possible values
`Size`	Memory used by the user session in bytes	Integer

Additional information:

- Operation: Threads Information
- Operation: JVM Parameters
- Operation: Processes Running
- Operation: User Sessions Memory
- Operation: Apps Status

10.22.16.5 Operation: Apps Status

The `Apps Status` operation outputs information about the failed and deployed X4 apps.

Examples

Output

The adapter generates an XML document as 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>
```

The following elements are included in the output:

Element	Description	Possible values
Number	Number of deployed X4 Apps	• Integer
App	Name of the X4 App file including the file extension	• Any string

For more information, see the section

- Operation: Threads Information
- Operation: JVM Parameters
- Operation: Processes Running
- Operation: User Sessions Memory
- Operation: Apps Status

10.22.17 Task Management

`Task Management`: Enables interface-based access to human tasks in Business Processes via Technical Processes. It forms an interface that makes modelled tasks and human interaction available on process level. This enables the connection of customer-specific Web Apps to processes modeled with BPMN or the direct connection of any third-party system to the central process control of the X4 BPMS.

For access via the adapter, the user logged on in X4 Web Apps is used.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `GetAllTasks`: Return all tasks for the logged-in user or group • `GetTask`: Return one task for the user logged-in • `EditTask`: Lock the tasks for other users and update task information • `CloseTask`: Complete the task with a professional status and continue the case • `DeclineTask`: Decline the task and return it to the owner • `ForwardTask`: Forward the task to a person or group • `CancelTask`: Cancel task processing and release the task for other users • `ResubmitTask`: Set task on resubmission

Parameters

`Adapter`	Main adapter class (Do not change!) **Possible values:** *de.softproject.x4.server.bpm.casemanagement.adapter.TaskManagement*: Main class (default)

`limit`	Page size for retrieving the task list The value is optional. If the parameter is not set, all tasks will be returned. **Possible values:** Any number *> 0*
`offset`	Offset for retrieving the task list The value is optional. If the parameter is not set, all tasks will be returned. **Possible values:** Any number *> 0*
`state`	*Required*. State filter for retrieving the task list **Possible values:** • *OPEN*: Open tasks • *IN_PROGRESS*: Tasks in progress • *CLOSED*: Closed tasks
`taskUuid`	Task UUID **Possible values:** Valid UUID, e.g. *443c498a-5868-4110-891f-42e5cb8fc24e*
`userForDebugging`	User to be used for debugging, if no logon exists in X4 Web Apps **Possible values:** Valid user login, e.g. *admin*
`dataformatVersion`	Version of the used data format **Possible values:** • `1.0`: Used data format until version 7.0 • `2.0`: Used data format since version 7.0, see sample inputs

Status values

`1`	The operation was executed successfully
`-1`	The operation could not be executed

The documentation of the ReST interface can be found at http://localhost:8080/X4/X4Api/.

Input

No input is required for following operations:

- `GetAllTasks`

- `GetTask`
- `CancelTask`

The following operations require an XML document with the following structure as input:

- Operation `EditTask`

  Example input for the EditTask operation

```
<?xml version="1.0" encoding="UTF-8"?>
<EditTask>
    <Offer> 150 Mio. €</Offer>
</EditTask>
```

- Operation `CloseTask`

  Example input for the CloseTask operation

```
<?xml version="1.0" encoding="UTF-8"?>
<CloseTask>
    <MarketValue>120 Mio. €</MarketValue>
    <Action>Declined</Action>
</CloseTask>
```

- Operation `DeclineTask`

  Example input for the DeclineTask operation

```
<?xml version="1.0" encoding="UTF-8"?>
<DeclineTask>
    <Comment>string</Comment>
</DeclineTask>
```

- Operation `ForwardTask`

  Example input for the ForwardTask operation

```
<?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>
```

Output

When the `CancelTask` operation is executed, the output is the input document.

The following operations output an XML document with the following structure:

- 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>
```

- Operations `GetTask`, `EditTask`, `CloseTask`, `DeclineTask` and `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`: Allows to retrieve information about the currently logged-in user, e.g. username, first and last name, set language, email address and groups this user is assigned to.

If a non-default realm is used, a mapper of type **Group Membership** must be set up in the client. The configuration can be found in the **Administration Guide**, in the Default Configuration section.

Properties

`Operation`	Operation executed by the adapter **Possible values:** • `GetUserIdentity`: Returns the information about the user.

Parameter

`Adapter`	Adapter main class (do not change!) **Possible values:** • *`de.softproject.x4.adapter.useridentity.UserIdentityAdapter`*: Main class (default)
`generateTestGroup`	Defines whether a test group entry should be generated when the adapter is executed in debug mode. Works only in debug mode. **Possible values:** • *`true`*: Generate an test group entry • *`false`*: Do not generate an test group entry
`testGroupName`	Technical name of the test group entry if `generateTestGroup` is set to *`true`*.
`testGroupDisplayName`	Displayed name of the test group entry if `generateTestGroup` is set to *`true`*.
`username`	Used username if `generateTestGroup` is set to *`true`*.
`lastname`	Last name used if `generateTestGroup` is set to *`true`*.
`firstname`	First name used if `generateTestGroup` is set to *`true`*.
`email`	Email address used if `generateTestGroup` is set to *`true`*.
`timezone`	User time zone according to ISO 8601 output in debug mode. **Possible values:** Time zone according to ISO 8601 duration format, e.g. *`PT5H`*

Status values

1	The adapter operation was executed successfully.
-1	An error occurred while executing the adapter.

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`: Allows to write the data in the data stream to the user session, read and list data from the user session and invalidate the user session.

Properties

`Operation`	Operation performed by the adapter **Possible values:** • `Store Content`: Writes the data in the data stream to the user session. • `Get Content`: Outputs the record from the user session specified in the `key` parameter. • `List Content`: Outputs a list of the data in the user session. • `Check User Session`: Checks if a user session exists and outputs the user session ID. • `Clear Session Data`: Deletes the data in the user session and invalidates the user session.

Parameter

`key`	For operation `Get User Session`: Used as filter. For Operation `Store User Session`: Used as the name with which the data in the data stream is stored in the user session. **Possible values:** • Any string

Status Values

`1`	The adapter operation was executed successfully.

`-1`	An error occurred while executing the adapter.
`0`	The adapter operation did not returned a result.

10.22.20 Web App Translation

`Web App Translation`: Returns the localized values for defined translation keys.

Properties

`Operation`	Defines the operation executed by the adapter. **Possible values:** • `GetTranslation`: Returns the localized values for defined translation keys.

Parameters

`Adapter`	Main class of the adapter (Do not change!) **Possible values:** • *`de.softproject.x4.server.webapp.translation.adapter.TranslationAdapter`*: Main class (default)
`language`	Language for which the values are read. **Possible values:** String with the defined language key from the web app definition
	Translation key/-s that is to be read. If several translation keys are to be read, these must be separated by `;`. **Possible values:** String with the defined translation key/-s from the language file.

Status values

`1`	The operation was executed successfully.
`-1`	An error occurred during the operation's execution.

Output

- Language file was read successfully:

```
<?xml version="1.0" encoding="UTF-8"?>
<Result>
    <Output key="loginLabel" value="User"/>
    <Output key="passwordLabel" value="Password"/>
</Result>
```

- No language was set:

```
<?xml version="1.0" encoding="UTF-8"?>
<Result>No language set</Result>
```

- No translation key was set:

```
<?xml version="1.0" encoding="UTF-8"?>
<Result>No keys set</Result>
```

> ⚠ If a translation key that is not defined is queried, the translation key itself is returned instead of the value. If a language file that does not exist is queried, the translation key itself is also returned.

10.22.21 X4 Management

`X4 Management`: Calls the methods of the *X4*-specific *JMX*-MBeans X4 Management. The methods *`SetAllOutOfService`* and *`SetAllInService`* are available.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *`SetAllOutOfService`*: Sets the property `In service` to *`false`* for *all* processes. This has the effect that no more processes are started. • *`SetAllInService`*: Sets the property `In service` to *`true`* for *all* processes. This has the effect that all processes can be (re)started.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`en.softproject.integration.adapter.x4management.X4ManagementCheckerAdapter`*: Main class (default)

Status values

1	The operation was executed successfully.
-1	The operation failed due to a technical error.

10.22.22 X4 Server Config Reader

`X4 Server Config Reader`: Outputs Java parameters from the *X4* startup script and *X4* configuration properties from the `X4config.xml` file in an XML document. The desired parameters are named in a specifically structured input XML document and are output again by the adapter supplemented by the corresponding values.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`GetProperties`*: Output configuration properties

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.ConfigReader`*: Main class (default)

Status values

1	The operation was executed successfully.
-1	The operation failed due to a technical error.

Input

This adapter expects an adapter-specific input XML structure. Any number of `<Param/>` elements can be entered; the `<X4Config/>` or `<JavaConfig/>` element can be omitted as required.

```
<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>
```

Output

This adapter adds the dynamically read values to the input XML document if possible, for example:

```
<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>
```

Example

The following is an example of an output XML document containing *X4* configuration key and values, and Java options and their values:

```
<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`: Collects results from process loops and saves them in a variable.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • `Collect`: Encode data Base64, see section Operation `Collect` • `Get`: Read the process variable's content, see section Operation `Get` • `GetAndRemove`: Read the process variable's content and remove the process variable subsequently, see section Operation `GetAndRemove`

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.merge.Collector :* Main class (default)
`name`	Name of the process variable
`mode`	Merge mode; Determines how the input is merged into the process variable. **Possible values:** • `BINARY`: The input is merged as Byte stream • `TEXT`: The input is merged as string • `XML`: The input is merged as XML document

Status values

`1`	The operation was executed successfully
`-1`	A technical problem occurred during the operation's execution
`0`	No process variable was found (for `Get` and `GetAndRemove`)

10.22.23.1 Operation `Collect`

Mode: `BINARY`

The following applies to the mode `BINARY`:

- The adapter input is interpreted as byte array and appended to the process variable's content that is interpreted as byte array.
- `NULL` is considered as byte array of length *0*.
- The result's content type is `application/octet-stream`.

- The result is written into the process variable and returned as adapter result.

Mode: TEXT

The following applies to the mode TEXT:

- The adapter input is interpreted as string (the encoding is considered therefore) and appended to the process variable's content that is interpreted as string (the encoding is used therefore).
- NULL is considered as string of length *0*.
- The result's content type is text/plain with the encoding UTF-8.
- The result is written into the process variable and returned as adapter result.

Mode: XML

The following applies to the mode XML:

- The adapter input is interpreted as XML and appended to the process variable's content that is interpreted as XML.
- The result's content type is text/xml with the encoding UTF-8.
- The result is written into the process variable and returned as adapter result.
- If the adapter input or the process variable's content can not be interpreted as XML (NULL is allowed), the adapter is terminated with the status *-1* and NULL as output. The process variable's content was not changed in this case.

It will be attached as follows:

- A new document with the root element Merge is created.
- If the process variable's content is not NULL, its content is merged as follows.
  If the process variable's root element is Merge, all child elements are taken over as child elements of the output Merge element. Otherwise, the root element is taken over as child of the output Merge element.
- Subsequently, the same treatment is applied to the adapter input.
  After a successful Collect in XML mode, the process variable contains in any case an XML document with a root element Merge. From the view of the Collect operation in XML mode, NULL (nothing) and the <Merge> XML document are equivalent.

10.22.23.2 Operation Get

The operation Get returns the process variable's content. If the variable is filled, the status *1* is returned; If it is empty, the status *0*. Thus, the operation Get behaves like the operation Test of the Variable component.

If the variable is empty, the neutral element is set into the variable and used as adapter output.

Mode	Corresponding neutral element
BINARY	Byte array of the length *0*
TEXT	String of the length *0*
XML	An XML document consisting exclusively of the root node Merge

10.22.23.3 Operation GetAndRemove

The operation GetAndRemove returns the process variable's content. If the variable is filled, the status *1* is returned; If it is empty, the status *0*. Thus, the operation GetAndRemoves combines the operations Remove and Test of the Variable component.

If the variable is empty, the neutral element is set into the variable and used as adapter output.

Mode	Corresponding neutral element
BINARY	Byte array of the length *0*
TEXT	String of the length *0*
XML	An XML document consisting exclusively of the root node Merge

The variable is empty after this operation.

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: Expands and collapses data segments in XML Documents based on the set parameters.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • Expand: Expand data segments embedded in an XML document. The data segments contain XML data as plain text or Base64-encoded strings and will be expanded and embedded into the input XML document. • Collapse: Collapse parts of an XML document into plain text or Base64-encoded strings. The collapsed data segments are embedded into the input XML document.

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.adapter.inlinecontentexpanderAdapter`*: Main class (default)
`select`	XPath expression to select from the input XML document **Possible values:** Any valid XPath expression (e.g. *`Root/Element[1]`*)
`Format`	Format of the data segments **Possible values:** • `BASE64`: Set the format to Base64-encoding (default) • `STRING`: Set the format to plain text
`Output`	Output format of the data segments **Possible values:** • `XML_ONLY`: Only processes valid XML structures. In case of invalid structures an error is output. (Default) • `XML_OR_STRING`: Processes XML structures if they are valid, otherwise the data is output as strings. • `STRING_ONLY`: Only processes strings.
`encoding`	Encoding of the data segments **Possible values:** • Any valid encoding (e.g. *`windows-1252`*) • `UTF-8`: UTF-8 (default)

Status values

`0`	The XPath expression doesn't reference any element; The input will be output without modification.
`1`	Adapter was executed successfully
`-1`	An error occurred during the adapter's execution. Depending on the operation, either the document is output without modification (`Collapse`) or the corresponding error message is output (`Expand`).

Input

This adapter accepts any valid XML document as input.

Sample input XML document

```
<?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>
```

Adapter parameters set for processing the input:

- Operation: *Collapse*
- select: *//Body*
- Format: *BASE64*
- Output: *XML_ONLY*
- encoding: *UTF-8*

Output

The adapter outputs an XML document with processed data embedded.

Sample 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`: Counts records in the input data determined by a delimiter. This adapter treats all input data as text and outputs an XML document that contains the number of records.

Properties

Operation	Determines which operation the adapter executes **Possible values:** *`Count`*: Count records in the input data

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *`de.softproject.integration.adapter.RecordCounter`*: Main class (default)
RecordID	Separator used (or escape sequence) **Possible values:** • Any character (e.g. `,`) • *`\n`*: Interpret line break as separator (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Output

The successfully executed *Record Counter* returns an XML document with the following pattern:

```
<RecordCount><!--Anzahl der Datensätze--></RecordCount>
```

10.23.3 XHTML Form Extractor

`XHTML Form Extractor`: Extracts form fields from XML-compatible (X)HTML documents to make the mapping of process data to form fields easier to understand.
The adapter expects an (X)HTML document as input (in any case a well-formed XML document!) and outputs an XML document with the structure of the extracted (X)HTML form. In addition, the adapter can use the `FillInFields` operation to fill the adapter-specific XML structure, which contains set form fields and, if applicable, modified properties for option fields, into an (X)HTML form template and output the finished (X)HTML form document.

> This adapter is targeted at developers of *Human Workflow* processes who want to fill existing (X)HTML form templates with data at runtime. For this purpose, the input data structure (e.g. the results of a database query) is linked to the fields of the (X)HTML form in an XSL mapping via the Mapping Editor.
> To make (X)HTML document structures clearer for mapping, the `XHTML Form Extractor Adapter` extracts all form elements so that only this simplified form structure and the data need to be linked in the Mapping Editor. For conversion to the XML form structure, the context menu function **Extract XHTML Form** for (X)HTML repository documents, is available in the *X4 Designer*.
> To convert the XML form structure back into a complete (X)HTML form after mapping, place the `XHTML Form Extractor Adapter` as the next process step, select the `FillInFields` operation and specify the URL to the (X)HTML form template in the adapter parameter `templateUrl`. When the process is executed, the modified form elements are merged with the (X)HTML form template and output at runtime, e.g. for display in web applications. Via attributes in the XML form structure, option fields can be (de)selected and (de)activated, and CSS direct formatting and CSS class assignments can be defined.

Properties

Operation	Determines which operation the adapter executes **Possible values:** • *`ExtractFields`*: Extract form fields from an input (X)HTML document (well-formed XML!). • *`FillInFields`*: Insert form fields from the adapter-specific XML structure into an XML-compliant (X)HTML document defined in parameter `templateUrl`; additional fields from the (X)HTML template document are taken over unchanged.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** en.softproject.integration.adapter.htmlformextractor.FormAdapter: Main class (default)
templateUrl	URL to the template document of the (X)HTML form (required for operation `FillInFields`). **Possible values:** Any valid URL (e.g. `xstore://Project/Folder/Form.xhtml`)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

Example

In this example, an HTML5 form (well-formed XML) serves as 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>
```

In Operation `ExtractFields`, the adapter converts the HTML5 form into the following XML structure:

```
<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>
```

The following XML structure has now been created via an XSL mapping, in which field values and the page title have been changed. Various properties for option fields have been set and an additional option field has been added.

```
<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>
```

With Operation `FillInFields`, the adapter converts the above modified XML structure back into a complete, XML-compliant HTML5 form:

```
<!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

With Operation `ExtractFields`, form fields can be extracted from an input (X)HTML document (well-formed XML!).

Input

For Operation `ExtractFields`, the adapter expects an (X)HTML document that has a well-formed XML structure (correct nesting, self-closing empty elements such as `<br/>` etc.) and whose form elements are unique via attribute `id` or `name`.

Output

With Operation `ExtractFields`, the adapter outputs an XML document with form data that you can use in the mapping as a template for the target structure.

10.23.3.2 Operation: FillInFields

Operation `FillInFields` can be used to insert form fields from the adapter-specific XML structure into an XML-compliant (X)HTML document defined in parameter `templateUrl`; additional fields from the (X)HTML template document are taken over unchanged.

Input

With operation `FillInFields`, the adapter expects an XML document that has a structure according to the following pattern and creates an XML element for all form fields (except checkboxes and radio buttons) that contains the field type and field name in the name.

In the XML structure, `<option/>` fields can be provided with CSS direct formatting (`cssStyle="..."`), CSS class assignments (`cssClass="..."`) and labels (`label="..."`) via specific attributes, and can be (de)selected (`selected="true/false"`), (de)activated (`disabled="true/false"`) and set to read-only (`readonly="true/false"`). Values of `<input/>` and `<option/>` can be set via an `attribute value="..."`.

```
<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>
```

Output

In the `FillInFields` operation, the adapter outputs an XML-compatible (X)HTML document that contains the field values (and, in the case of radio buttons, possibly modified properties) that were

defined in the input document and whose structure corresponds to the template defined in the `templateUrl` parameter. Checkboxes and radio buttons from the template are transferred unchanged to the result document.

10.23.4 XML Checker

`XML Checker`: Checks if the input document is an XML document, and outputs the document with a corresponding status.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** *checkXML*: Check the input document

Parameter

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.typechecker.TypeCheckerAdapter*: Main class (Default)

Status values

0	The checked input document is no XML document.
1	The checked input document is a well-formed XML document.

10.23.5 XML Element Wrapper

`XML Element Wrapper`: Wraps the input data with a specific XML root element and outputs a UTF-8 encoded XML document.

Properties

Operation	Determines which operation the adapter executes **Possible values**: *Wrap*: Output input data as XML document with root element from parameter `name`

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *en.softproject.integration.adapter.elementwrapper.ElementWrapperAdapter*: Main class (default)

name	Name of the root element **Possible values:** • Any valid XML element name • *`Data`*: Root element *`Data`* (default)

Status values

`1`	The operation was executed successfully.
`-1`	The operation failed due to a technical error.

10.23.6 XML Formatter

`XML Formatter`: Provides formatting functions for XML documents, e.g. indents and line breaks.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • *`PrettyPrint`*: Format XML document

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.xmlformatter.XMLFormatterAdapter*: Main class (default)
indentSize	Number of indents within the XML document **Possible values:** Any integer value, e.g. *4*

Status values

1 (successful)	The request was successful and a result was returned
-1 (failed)	The request failed due to a technical error

Input

The adapter expects an unformatted XML document as input.

Output

The adapter outputs a formatted XML document.

10.23.7 XML Iterator

`XML Iterator`: Reads an XML file from any valid URL incrementally into a Technical Process and outputs for each iteration a child element (including all child nodes) of the root element as XML document

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • *`Iterate`*: Read child node of the root element • *`Clear`*: Reset iterator of an *XML Iterator*, which has the same ID as the respective *XML Iterator* with the operation *`Iterate`* (see parameter `id`) ⚠ After executing the *`Iterate`* operation, execute the *`Clear`* operation to avoid possible resource leaks.

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.adapter.xmliterator.XmlIterator`*: Main class (default)
url	URL of the XML document to be read **Possible values:** • Any valid URL (e.g. *`xstore://Project/Folder/File.xml`*) • URL to a UNC network drive (e.g. *`file://///storage/share_data/Folder/File.xml`*)

id	Unique ID of the XML Iterator component within the process context respectively the process instance (if you want to use several *XML Iterators* iterating independently within the same process context) **Possible values:** • *0*: Use current action ID (number of the process step of the *XML Iterator* within the current process) as ID; Please note that an *XML Iterator* may have the same action ID in a sub-process (default) • Any integer number except *0*, to uniquely identify the respective *XML Iterator* component within the process context; Recommendation: use a value *>1000*, to avoid that an already existing action ID is taken "accidentally"

State values

-1 (error)	An error occurred during the adapter's execution (see *X4 Server* log).
0 (empty)	The adapter doesn't output data (anymore). This status can be used as stop criterion of a loop within the Technical Process and it indicates that no further data set is available.
1 (successful)	The adapter outputs a result. This status can be used as condition to enter a loop within the Technical Process and it indicates that another data set is available.

10.23.8 XML Text Extractor

`XML Text Extractor`: Extracts the contents of a node within an XML document as a text document. The node is targeted using an XPath expression.

> **Extractable elements**
> You can extract only elements of type CDATA, attributes or text.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Extract`: Extract the content of a node

Parameters

`Adapter`	Main adapter class (Do not change!) **Possible values:** *`de.softproject.integration.adapter.xmltextextractor.XMLTextExtractor`*: Hauptklasse (default)
`xpath`	XPath expression to select from the input XML document **Possible values:** Any valid XPath expression (e.g. *`Root/Element[1]`*). If the XPath expression targets more than one element, an error is issued. To extract more than one element, combine the adapter with the XPath Iterator adapter.

Status values

`0`	The adapter doesn't output data (anymore). This status can be used as stop criterion of a loop within a Technical Process and indicates that no further data row is available.
`1`	The adapter outputs a result document. This status can be used as condition to enter a loop within a Technical Process and indicates that a further data row is available.
`-1`	An error occurred during the adapter's execution (for details see server log).

Input

The adapter expects any valid XML document as input.

Example input XML document

```
<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>
```

Set adapter parameters:

- `xpath`: *`//address[1]/name`*

Output

The adapter outputs a text document with processed data.

Example output

```
John Doe
```

10.23.9 XML Validation

`XML Validation`: Validates an input XML document against an XML schema or a Relax NG schema. If the validated XML document is valid, the adapter outputs the document without modifications; If the document is invalid, an XML document with the corresponding error message will be output.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** `Validate`: Validate XML document

Parameters

`Adapter`	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.xmlvalidator.XMLValidator*: Main class (Default)
`SchemaUrl`	Path to the schema definition (XML schema or Relax NG) **Possible values:** Any `xstore://` URL (e.g. *`xstore://Project/Folder/Schema.xsd`*)
`Verbose`	Specification, whether the result document should contain detailed information on the validation error **Possible values:** • `true`: The result document contains detailed information on the validation error (default) • `false`: The adapter stops validation after the first error and doesn't output any information on the error
`xsdVersion`	XSD version to be used to validate XML documentsl **Possible values:** • `1.0`: XSD version 1.0 is used • `1.1`: XSD version 1.1 is used

Status values

`-1 (error)`	An error occurred during the adapter's execution. For more detailed information see the server log.
`1 (successful)`	The XML document is valid according to the specified schema; The function adapter has output the verified input XML document unmodified.
`2`	The XML document is invalid according to the specified schema; The function adapter has output an error message (see Output).

Output

The function adapter outputs a UTF-8-encoded XML document. If the validation was successful, the adapter will output the input document without modifications. If the document is invalid, an XML document with the corresponding error message will be output. An XML element `<Error/>` containing the detailed error description will be created for each error, e.g.:

```
<?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`: Selects a node set, a node or node content using an XPath expression in an input XML document, and outputs a node set as XML document or values as text document. The adapter can either iterate through all matching nodes and output the results successively (`ForEach` loop) or it can output the result of the first match.

> If you define several *XPath Iterators* within one process context, it is required to assign an explicit ID for the corresponding adapter within the parameter `Id`, e.g. *`1001`*, *`1002`* etc. Thus, it is ensured that only the *XPath Iterator* with the corresponding ID will be affected (see operation `Clear`) when selecting or resetting the iterator in the `forEach` mode.
> If the default value of the parameter `Id` remains on *`0`*, the ID is equivalent to the process step number (action ID) in the current process. This ID may not be unique, if an *XPath Iterator* with the same action ID is used in a subprocess.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *`Select`*: Apply XPath expression to the input XML document and output a result document (consider parameter `Id`) • *`Clear`*: Reset an *XPath Iterator's* iterator having the same ID as the *XPath Iterator* using the operation `Select` (see parameter `Id`)

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.adapter.xpathselector.XPathSelector`*: Main class (default)
XPath	XPath expression applied to the input XML document **Possible values:** Any XPath expression (e.g. */`Element[@att="Value"]/text()`*)

forEach	Select the first node or all matching nodes **Possible values:** • `true`: Iterate through all selected nodes and output a document in each case, useful in combination with a `Condition` process component (default) • `false`: Output first match
Id	Unique ID of the *XPath Iterator* component within the process context or within the process instance (when using several independently iterating *XPath Iterators* within the same process context) **Possible values:** • `0`: Use current action ID (process step number of the *XPath Iterator* in the current process) as ID; Please note that an *XPath Iterator* in a subprocess could have the same action ID (default) • Any integer number except `0` in order to identify the *XPath Iterator* component within the process context; Recommendation: use a value `>1000` in order to avoid catching an already existing action ID

Status values

-1 (error)	Error when executing the adapter (for details, see the server log)
0 (empty)	The adapter doesn't output data (anymore). In the iterator mode, this status can be used as stop criterion of a loop within the Technical Process. It indicates that no further data row is available.
1 (successful)	The adapter outputs a result. In the iterator mode, this status can be used as condition to enter a loop within the Technical Process. It indicates that further data row is available.

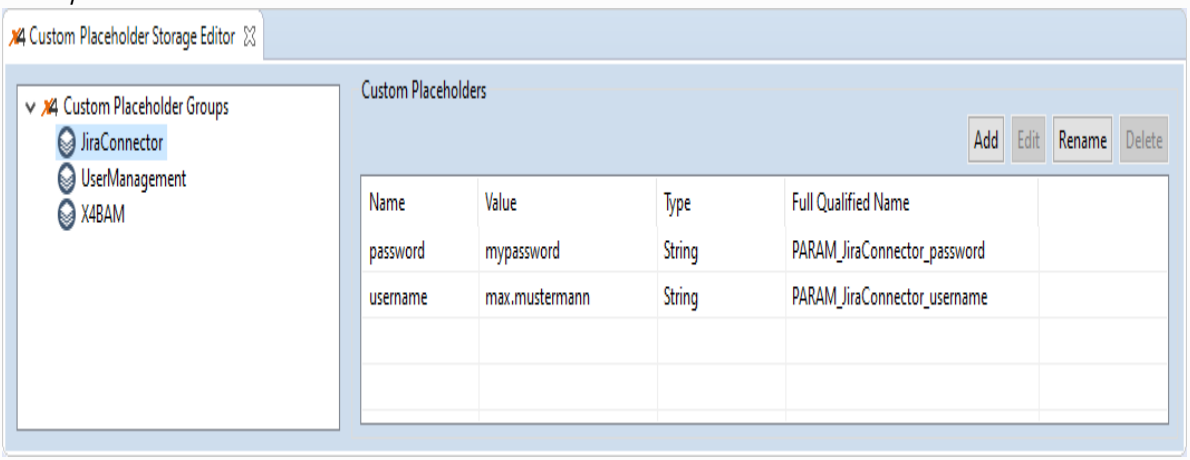
10.24 Deprecated adapters

This section lists all deprecated X4 Adapters. Deprecated X4 adapters are no longer supported and will not be updated or modified.

10.24.1 Atlassian Jira Connector

`Atlassian Jira Connector`: Connects processes with Atlassian Jira.

Please note that in you first need to create a placeholder group with the appropriate Jira credentials to use the adapter. You can do this either manually in the Global Parameter Configuration or via an integration process using the adapter Global Parameter Management.
Example:



Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Get Issue`: Get an issue from a specific project • `Get all Issues`: Get all issues from a specific project • `Create Issue`: Create issue in a specific project • `Edit Issue`: Edit available fields in an existing issue • `Add Attachment`: Add an attachment to a specific issue • `Remove Attachment`: Delete attachment of an issue by global ID • `Get all Comments`: Get all comments from a specific issue • `Get Comment`: Get a specific comment from an issue • `Add Comment`: Add a new comment to an issue • `Edit Comment`: Edit a specific comment • `Search`: Search by parameters • `Execute Transition`: Perform status transition for a specific issue • `Get Transition`: Get status of a specific issue

Parameters

`Adapter`	Main adapter class (Do not change!) **Possible values:** *`de.softproject.integration.nativeadapter.NativeAdapter`* : Main class (default)

`Repository`	Virtual repository of the adapter project. The value cannot be changed.
`host`	URL of the Jira server **Possible values:** Valid URL, e.g. *`jira.example.com`*
`port`	Port of the ReST API **Possible values:** • Positive integer • *`443`*: Default port
`key`	Project key with or without issue ID **Possible values:** Project key, e.g. `XVIER`, *`XVIER-12345`*
`secure`	Defines if the communication should take place via HTTPS or not **Possible values:** `true` (default) / `false`
`username`	Jira user to be used for authentification
`password`	Password of the Jira user to be used for authentication
`token`	Token used for the authentication
`query`	*Used with operation* `Get all Issues` ; Search criteria as a "Jira Query Language" expression. **Possible values:** Valid search criteria, e.g. • `project=XVIER and Rang < 999` • `project=ITSERVICE and reporter=max.mustermann`
`offset`	Offset at which the enumeration of operations starts
`count`	Maximum number of returned issues
`id`	ID that specifies comments or attachments

Status values

`1`	The adapter operation was executed successfully.
`-1`	An error occurred during the adapter's execution.

Input

The adapter expects different input structures depending on the selected operation. For more information, see the following subchapters.

Output

The adapter outputs different structures depending on the selected operation. For more information, see the following subchapters.

10.24.1.1 Get Issue

Input

- No input is required for this operation.
- This operation requires the `host`, `key`, `username` and `password` or `token` parameters.
- The project key with the task ID is specified in the `key` parameter.

Output

- If the operation was performed successfully, then the adapter outputs the issue with all the fields filled in.

Get Issue

```
<JSON>
    <expand>renderedFields,names,schema,operations,editmeta,changelog,versionedRepres
entations</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.Summ
aryBean@405cbc05[summary={pullrequest=com.atlassian.jira.plugin.devstatus.rest.Summar
yItemBean@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

- No input is required for this operation.
- This operation requires the `host`, `key` or `query`, `username` and `password` or `token` parameters.
- The project key with the task ID is specified in the `key` parameter. More search criteria can be defined in the `query` parameter.

Output

- If the operation was performed successfully, then the adapter outputs all issues with all the fields filled in.

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.rest.SummaryBean@31b2d52c[summary={pullrequest=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@24aa33fa[overall=PullRequestOverallBean{stateCount=2, state='MERGED', details=PullRequestOverallDetails{openCount=0, mergedCount=2, declinedCount=0}},byInstanceType={stash=com.atlassian.jira.plugin.devstatus.summary.beans.ObjectByInstanceTypeBean@7e649bc6[count=2,name=Bitbucket Server]}], build=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@66337825[overall=com.atlassian.jira.plugin.devstatus.summary.beans.BuildOverallBean@228f2b11[failedBuildCount=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[topEnvironments=[],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.ObjectByInstanceTypeBean@357dd6cf[count=5,name=Bitbucket Server]}], branch=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@dd72830[overall=com.atlassian.jira.plugin.devstatus.summary.beans.BranchOverallBean@51e0f1c4[count=0,lastUpdated=<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

- This operation requires the `host`, `key`, `username` and `password` or `token` parameters.

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

- This operation requires the `host`, `key`, `username` and `password` or `token` parameters.

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

- This operation requires the `host`, `key`, `username` and `password` or `token` parameters.
- The file to be attached must be Base64 encoded.
- Several files can also be attached at the same time.

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+SGF
sbG8gV2VsdCE8L3Rlc3Q+DQo8L3Jvb3Q+</fileData>
        </attachment>
    </attachments>
</RootElement>
```

Output

- If the operation was performed successfully, then the adapter outputs the global Jira ID of the attachment.

```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

- No input is required for this operation.
- This operation requires the `host`, `username`, `password` or `token` and `id` parameters.
- The ID of the attachment is specified in the `id` parameter.

Output

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Message>The operation was successful!</Message>
```

10.24.1.7 Get all Comments

Input

- No input is required for this operation.
- This operation requires the `host`, `key`, `username` and `password` or `token` parameters.
- The project key with the task ID is specified in the `key` parameter.

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

- No input is required for this operation.
- This operation requires the `host`, `key`, `username`, `password` or `token` and `id` parameters.
- The project key with the task ID is specified in the `key` parameter, the corresponding ID of the comment in the `id` parameter.

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>2021-02-05T14:42:33.967+0100</updated>
</JSON>
```

10.24.1.9 Add Comment

Input

- This operation requires the `host`, `key`, `username` and `password` or `token` parameters.

```
<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

- This operation requires the `host`, `key`, `username`, `password` or `token` and `id` parameters.

- The project key with the task ID is specified in the `key` parameter, the corresponding ID of the comment in the `id` parameter.

```
<?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
<?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

- No input is required for this operation.
- This operation requires the `host`, `username`, `password` or `token` and `query` parameters.
- More search criteria can be defined in the `query` parameter.

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

- This operation requires the `host`, `key`, `username` and `password` or `token` parameters.

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

If the operation was executed successfully, the adapter outputs `OK`.

Execute Transition

```
<Ok>
    <Message>Executing transition on issue was successful</Message>
</Ok>
```

Otherwise, the status `-1` is output. If it is possible to retrieve possible transitions, they will be output (in case of wrong transition ID, this will help to find the correct ID).

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

- No input is required for this operation.
- This operation requires the `host`, `key`, `username` and `password` or `token` parameters.
- The project key with the task ID is specified in the `key` parameter.

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`: Enables a low level access to the Atlassian Jira ReST API.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Execute`: Execute the adapter with the configured parameters

Parameters

`Adapter`	Main adapter class (Do not change!) **Possible values:** *`de.softproject.integration.nativeadapter.NativeAdapter`* : Main class (default)
`Repository`	Virtual repository of the adapter project. The value cannot be changed.
`host`	*Required*. URL of the Jira server **Possible values:** Valid URL, e.g. *`jira.example.com`*
`port`	Port of the ReST API **Possible values:** • Positive integer • *`443`* : Default port
`path`	*Required*. Resource path for the request **Possible values:** URL part in jira (For more information seehttps://docs.atlassian.com/software/jira/docs/api/REST/7.13.0/#api/2), e.g. */rest/api/2/{issueIdOrKey}*
`secure`	Defines if the communication should take place via HTTPS or not **Possible values:** `true` (default) / `false`
`username`	*Required*. Jira user used for authentication
`password`	*Required if `token` is not used*. Password of the Jira user to be used for authentication
`token`	*Required if `password` is not used*. Token used for authentication. The token can be used instead of a password.
`method`	*Required*. HTTP method used for the request **Possible values:** • `Get` (default) • `Post` • `Put` • `Delete` • `Patch`
`query`	URL encoded query, e.g. *query="project={projectKey} ORDER BY createdDate ASC"*

Status values

1	The adapter operation was executed successfully.
-1	An error occurred during the adapter's execution.

Input

The adapter expects input in XML format. The root element must contain the namespace definition `xmlns:j=http://www.softproject.de/schemas/2020/json-converter`. The exact structure can be found at https://docs.atlassian.com/software/jira/docs/api/REST/7.13.0/#api/2.

Output

The adapter returns a document with the structure described at https://docs.atlassian.com/software/jira/docs/api/REST/7.13.0/#api/2, but in XML format. If there is no content that can be output, then the message `The operation was successful!` is output.

10.24.3 Global Parameter Initializer (previously: Custom Placeholder Initializer) (deprecated)

`Global Parameter Initializer`: Manages global parameters within the current process. The loading and unloading of parameters from the process context can be controlled by means of adapter parameters or an XML document.

Properties

`Operation`	Defines the operation executed by the adapter **Possible values:** • `Load`: Loads the global parameters defined in the adapter parameter `groups` to the process context (*input XML = output XML*) • `Unload`: Unloads the global parameters defined in the adapter parameter `groups` from the process context (*input XML = output XML*) • `LoadFromXML`: Loads the global parameters from the XML document to the process context • `UnloadFromXML`: Unloads the global parameters from the XML document from the process context • `Export`: Outputs all currently set global parameters to the process context (doesn't need any input document)

Parameter

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.globalparameter.GlobalParameterInitializerAdapter*: Main class (default)

groups	List of the global parameter groups to be loaded from the global parameter storage (**Global Parameter Configuration**; previously: **Custom Placeholder Storage Editor**). Values can be separated by comma or semicolon. *Example:* `DEFAULT, Prod, Dev, Test`

Status values

1 (successful)	The operation was successful
0 (empty)	No global parameters were loaded to the process context
-1 (failed)	The operation failed due to a technical error

Input

The operations `LoadFromXML` and `UnloadFromXML` expect the following XML format:

LoadFromXML

```
<?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEV">
        <Key name="message">Hello developer!</Key>
    </Group>
    <Group name="DEFAULT">
        <Key name="message">Hello world!</Key>
    </Group>
</Placeholder>
```

The two global parameter groups `DEFAULT` and `DEV` with the corresponding global parameter *`message`* are stored in the parameter context.

UnloadFromXML

```
<?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEV" />
    <Group name="DEFAULT">
        <Key name="message" />
    </Group>
</Placeholder>
```

All global parameters of the group `DEV` and the global parameter *`message`* from the *`DEFAULT`* global parameter group are removed from the process context.

Output

The operation `Get` outputs all current global parameters within the process context. The operations `LoadFromXML` and `UnloadFromXML` output the processed elements.

```
<?xml version="1.0" encoding="UTF-8"?>
<Placeholder>
    <Group name="DEV">
        <Key name="message">Hello developer!</Key>
    </Group>
    <Group name="DEFAULT">
        <Key name="message">Hello world!</Key>
    </Group>
</Placeholder>
```

10.24.4 MongoDB Connector

10.24.4.1 About the adapter

`MongoDB Connector`: Connects to a MongoDB and executes one or more database statements passed as an input XML document with adapter-specific structures.

Parameter

Adapter	Main class of the adapter (do not change!) **Possible values:** *de.softproject.integration.adapter.mongodb.MongoAdapter*: Main class (default)
ConnectionName	Alias/reference name for the connection; Corresponds to the `Connection name` from the connection configuration, see Configure connection. **Possible values:** Name of the connection (e.g. *#default* (default)).

ReadPreference	Read-preference mode; this parameter defines how reads are processed **Possible values:** • `PRIMARY`: All operations read from the `primary` node of the current replica set. • `PRIMARY_PREFERRED`: In most cases all operations read from the `primary` node, but if it is not available, read from the `secondary` nodes (default). • `SECONDARY`: All operations read from the `secondary` nodes of the current replica set. • `SECONDARY_PREFERRED`: In most cases all operations read from the `secondary` nodes, but if these are not available, read from the `primary` node. • `NEAREST`: The operations read from the node of the replica set with the lowest network latency, regardless of the node type. Additional information on read-preference modes is also available at: • http://docs.mongodb.org/manual/core/read-preference/#read-preference-modes • http://api.mongodb.org/java/current/com/mongodb/ReadPreference.html#method_summary

WriteConcern	Indicates whether a `write` operation was successfully executed. Write Concerns allow to detect input errors or invalid `mongod` instances. **Possible values:** • `ERRORS_IGNORED`: No exceptions are thrown, not even in case of network problems. ⓘ This write concern is and has no replacement. The write concern `UNACKNOWLEDGED` is the most comparable. However, all exceptions of the type *MongoSocketException* must be caught via *catch and ignore.* • `ACKNOWLEDGED`: The operation waits for confirmation from the primary server before issuing a result; exceptions are thrown for both network problems and server errors. • `UNACKNOWLEDGED`: The operation returns the result as soon as the message is written to the socket. Exceptions are thrown only for network problems, but not for server errors. • `FSYNCED`: The operation waits until the server has ensured that the data has arrived on disk. Exceptions are thrown in the event of both network problems and server errors. • `JOURNALED`: The operation waits until the server commits the data to the journal files. Exceptions are thrown in case of network problems as well as server errors. ⓘ **Notice:** • Journaling must be enabled to use this write concern. • With this write concern, operations must wait until the next journal commit. To reduce the waiting time for these operations, *MongoDB* increases the commit frequency, see also commitIntervalMs. • `REPLICA_ACKNOWLEDGED`: The operation waits for at least two servers before executing; exceptions are thrown for both network problems and server errors. • `NONE`: No exceptions are thrown, not even in case of network problems. ⓘ This write concern is obsolete and has been replaced by the write concern `ERRORS_IGNORED` and may be deprecated in future releases.

- `NORMAL`: The operation prints the result as soon as the message is written to the socket; exceptions are only thrown in case of network problems, but not in case of server errors.

  > This write concern has been replaced by the write concern `UNACKNOWLEDGED` and may be deprecated in future releases .

- `SAFE`: The operation waits for confirmation from the primary server before issuing a result; exceptions are thrown for both network problems and server errors.

  > This write concern has been replaced by the write concern `ACKNOWLEDGED` and may be deprecated in future releases.

- `MAJORITY`: The operation waits for the majority of servers before executing; exceptions are thrown for both network problems and server errors.
- `FSYNC_SAFE`: The operation waits until the server has ensured that the data has arrived on disk; exceptions are thrown in the event of both network problems and server errors.

  > This write concern has been replaced by the write concern `FSYNCED` and may be deprecated in future releases.

- `JOURNAL_SAFE`: The operation waits until the server commits the data to the journal files; exceptions are thrown in case of network problems as well as server errors.

  > This write concern has been replaced by the `JOURNALED` write concern and may be deprecated in future releases .

- `REPLICAS_SAFE`: The operation waits for at least two servers before it is executed; exceptions are thrown in case of network problems as well as server errors.

  > This write concern has been replaced by the `REPLICA_ACKNOWLEDGED` write concern and may be deprecated in future releases.

Additional information on Write Concerns is also available at:

- 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

Status values

`1`	The operation was executed successfully.
`0`	The operation was executed successfully, but without any result.
`-1`	The operation failed due to a technical error.

10.24.4.2 Configuring connection

The connection to *MongoDB* is configured in `mongodb-connections.xml` inside the X4 installation folder, e.g. `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>
```

Explanation:

- `/MongoDB/Connection`: Represents a named connection; May exist multiple times.
- `/MongoDB/Connection/@name`: Alias/reference name for the connection.
- `/MongoDB/Connection/@db`: Name of the database to which the connection should be made, i.e. databases must always be created via external tools (e.g. *MongoShell*).
- `/MongoDB/Connection/Server`: Represents a server entry for a connection; may exist multiple times per connection to address a cluster.
- `/MongoDB/Connection/Server/@host`: IP/DNS name of the server to connect to (`default: 127.0.0.1`)
- `/MongoDB/Connection/Server/@port`: Port of the server to connect to (`default: 27017`)
- `/MongoDB/Connection/Credential`: credentials for a connection; may exist multiple times per connection.
- `/MongoDB/Connection/Credential/@user`: User name of the login

- `/MongoDB/Connection/Credential/@password`: Login password
- `/MongoDB/Connection/Credential/@source`: Origin of credentials (usage currently unclear)

> **Notice:**
> If a connection has no server elements, it connects against `localhost:27017` without authentication data, regardless of whether credential elements exist or not.

10.24.4.3 Coding of a document structure

The *JSON/Mongo* documents are translated into their XML equivalent according to simple rules:

- Documents and objects/substructures are represented by `<Object>`. The key-value entries of an object are mapped as child elements of the `Object` element.
- Arrays are represented by `<Array>`. The array entries are mapped as child elements of the `Array` element.
- If it is not an array or object, it is represented by `<Value>`.
- The data type of a Value is recorded in the `type` attribute on the `Value` element (see Supported data types). If no data type is specified, `String` is assumed.
- The key-value entries of an object or a substructure get the key set as `name` attributes on the element of the value element.

Invariants:

- For each `{` in the JSON, an `Object` element is created in the XML. (Exception: the document root is mapped by other elements in commands).
- For each `[` in the JSON, an `Array` element is created in the XML.
- For each `:` in the JSON (which is not part of a string literal) a `name` attribute is created which contains the value before the `:` as content.
- All child elements of an `Object` element have a `name` attribute.
- Child elements of an `Array` element do not have a `name` attribute
- `Value` elements do not have a child element.
- `Text` nodes exist only as children of `Value` elements (exception: formatting whitespace).
- `type` attributes exist only on `Value` elements (but not every `Value` element must have a `type` attribute).

Sample

```
<Object>
    <Value name="some" type="String">thing</Value>
</Object>
```

> The document root can also be replaced by other elements to avoid the creation of an `Object` element. In this case, an `Object` element would always be necessary there.

10.24.4.4 Supported data types

The adapter supports the following data types:

- `Null`
- `ObjectId`
- `Boolean`
- `Long`
- `Int`
- `Double`
- `String`
- `DateTime`
- `Integer`: String is read as `BigInteger` and then stored as `Long`
- `Decimal`: String is read as `BigDecimal` and then stored as `Double`
- `Numeric`: If the string contains a `.` it will be read as `Decimal`, otherwise as `Integer`
- `RegEx`: Regular expression according to http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html

> This `Value` element also knows the `flags` attribute, which is a string of the following characters and sets the corresponding option on the pattern:
>
> - `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

The *MongoDB Adapter* expects specific input XML structures. These input XML structures contain a root with several commands underneath. Likewise, the `resulttag` attribute is also supported.

Input structure

```
<!-- 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>
```

This is a complete and executable example. All subsequent examples are merely snippets that must be inserted into a complete input structure with an enclosing root element in order to execute.

Supported commands

- copy (no MongoDB command)
- find
- insert
- remove
- save
- update

Command copy

With copy all child elements of the copy command are copied to the output.

Input

The copy command has the following syntax:

```
<copy resulttag="output">
    <Child1/>
    <Child2>
        <Grandchild1/>
        <Grandchild2/>
    </Child2>
</copy>
```

Output

This command outputs the following XML structure:

```
<output>
    <Child1/>
    <Child2>
        <Grandchild1/>
        <Grandchild2/>
    </Child2>
</output>
```

Command `insert`

With `insert` a document is inserted into the collection. The `insert` element is also the document root.

- Please note that `Batchinsert` with `insert` is not supported.
- More information about this command can also be found at http://docs.mongodb.org/manual/reference/method/db.collection.insert/#db.collection.insert.

Input

The `insert` command has the following syntax:

```
<insert collection="foo">
    <Value name="bar" type="String">baz</Value>
</insert>
```

Output

This command outputs a `WriteResult` object containing the status of the operation.

Command `save`

With `save` a new document is created or an existing document is updated according to the given data. The `save` element is also the document root.

More information about this command can also be found at http://docs.mongodb.org/manual/reference/method/db.collection.save/#db.collection.save.

Input

The `save` command has the following syntax:

```
<save collection="foo">
    <Value name="bar" type="String">baz</Value>
</save>
```

Output

This command outputs a

`WriteResult`

object containing the status of the operation.

Command `update`

With `update` a document in a collection is modified. Depending on the

`update`

parameter, the method can modify specific fields of an existing document or a complete document.

> More information about this command can also be found at http://docs.mongodb.org/manual/reference/method/db.collection.update/#db.collection.update.

Input

The

`update` command has the following syntax:

```
<update collection="foo">
    <query/>
    <update upsert="false" multi="false"/>
</update>
```

Explanation:

- The `query` element contains the query/selection part of the `update` and is also the document root for the query.
- The `update` element contains the change part of the update, and is also document root for the change.
- The `upsert` and `multi` options are specified on the inner `update` element.

Output

This command outputs a

`WriteResult` object containing the status of the operation.

Command `remove`

With `remove` a document is removed from a collection.

More information about this command can also be found at http://docs.mongodb.org/manual/reference/method/db.collection.remove/#db.collection.remove.

Input

The `remove` command has the following syntax:

```
<remove collection="foo">
    <query/>
</remove>
```

Explanation:

- The `query` element contains the query/selection part of the `remove` and is also the document root for the query.

Note that the `justOne` option is not supported by the Java driver.

Output

This command outputs a `WriteResult` object containing the status of the operation.

Command `find`

With `find` a document is selected from a collection.

More information about this command can also be found at http://docs.mongodb.org/manual/reference/method/db.collection.find/#db.collection.find.

Input

The `find` command has the following syntax:

```
<find collection="foo">
<--! Optional elements -->
  <query/>
  <projection/>
    <sort/>
    <hint/>
    <min/>
    <max/>
    <special/>
    <cursor skip="0" limit="0" batchSize="0"/>
</find>
```

Explanation:

- The `query` element contains the query/selection part of `find` and is also the document root for the query.
- The `projection` element contains the field selection of `find` and is also document root for the selection.

The other optional elements control the behavior of the result cursor:

- `sort` (Document root), see also http://docs.mongodb.org/manual/reference/method/cursor.sort/#cursor.sort
- `hint` (Document root), see also http://docs.mongodb.org/manual/reference/method/cursor.hint/#cursor.hint
- `min` (Document root), see also http://docs.mongodb.org/manual/reference/method/cursor.min/#cursor.min
- `max` (Document root), see also http://docs.mongodb.org/manual/reference/method/cursor.max/#cursor.max
- `special` (quasi-document root), see also http://api.mongodb.org/java/current/com/mongodb/DBCursor.html#addSpecial(java.lang.String,%20java.lang.Object)

  > For each child node in the root of the `special` document, the `addSpecial` method is called.

- `cursor` (no body) is only the carrier for the following three attributes:
  - `skip`, see also http://docs.mongodb.org/manual/reference/method/cursor.skip/#cursor-skip
  - `limit`, see also http://docs.mongodb.org/manual/reference/method/cursor.limit/#cursor.limit
  - `batchSize`, see also http://docs.mongodb.org/manual/reference/method/cursor.batchSize/#cursor.batchSize

Output
This command outputs a Cursor pointing to the documents that meet the `query` criteria.

10.24.4.6 Samples

The following examples show Mongo expressions and their respective equivalents in XML.

Examples of document structure:

JSON/Mongo structure	Corresponding XML structure
`{}`	`<Object/>`

JSON/Mongo structure	Corresponding XML structure

`{ 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>`

`{ $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>`

Call examples:

Mongo call	Corresponding XML structure
`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>`
`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>`

Mongo call	Corresponding XML structure

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>
db.inventory.update({ type : "book" }, { $inc : { qty : -1 } }, { multi: true })	<update collection="inventory"> <query> <Value name="type" type="String">boo k</Value> </query> <update multi="true"> <Object name="$inc"> <Value name="qty" type="Integer"> -1</Value> </Object> </update> </update>

Mongo call	Corresponding XML structure
`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>`

`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>`
	or
	`<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 call	Corresponding XML structure
`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>` or `<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 (deprecated)

`Repository Explorer`: Lists files and folders of a specific X4 Repository path (e.g. in folder `C:\X4\X4DB`) or of the complete repository.

The adapter outputs an XML document that lists the file and folder names. Depending on the selected operation, folder contents can be listed recursively (including subfolders). Optionally, the content type can be discovered; for XML files the root element of each file will be included.

> ⚠ The *Repository Explorer* is discontinued with version 6.2.0 of the X4 BPMS and will not be supported in future versions of the X4 BPMS. Please use the Project Access or Projects adapters instead.

> ⓘ When identifying the content type of repository files via the parameter `DiscoverContent`, it may take some time to process folder structures with numerous files!

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • *`List`*: List file and folder names that are located within `StartPath` • *`DeepListFlat`*: List file and folder names including subfolders that are located within `StartPath` • *`DeepListNested`*: Equivalent to `DeepListFlat`, but the result XML document displays a hierarchical folder structure • *`DeepListFlatFromRoot`*: List file and folder names of the whole *X4 Repository* (`X4DB`) including all subfolders • *`DeepListNestedFromRoot`*: Equivalent to `DeepListFlatFromRoot`; but the result XML document displays a hierarchical folder structure

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.adapter.repository.RepositoryAdapter`*: Main class (default)
StartPath	Repository path **Possible values:** Any path in the repository folder, including the user collection (e.g. *`1/Project/Folder`*)

DiscoverContent	Check the content type of repository files (XML or non-XML) **Possible values:** • *true*: Check the content type of repository files, and add to each element `File` within the result XML document a child element with the identified content type (affects performance!) • *false*: No content check (default)

Input

This adapter expects no input.

Output

Depending on the selected operation, the adapter outputs an XML document with the following pattern. Each folder will be output in an element `<Directory>`, files will be output in an element `<File>`:

```
<Repository repositoryPath="Absolute repository path within the file system">

   <!-- For Folders: -->
   <Directory name="Folder ername"
      path="Repository path of the folder including the User Collection"
      fullpath="Absolute repository path of the folder in the file system" />

   <!-- For Files: -->
   <File name="File name"
      path="Repository path of the file including User Collection"
      fullpath="Absolute repository path of the file in the file system">

      <!-- if 'DiscoverContent' has been activated: -->
      <Content type: XML or non XML>
         <!-- If Content Type is XML: -->
         <Root element name />
      </Content type: XML or non XML>
   </File>
</Repository>
```

Example

The following example outputs the hierarchical structure of a folder `Folder` using the operation `DeepListNested`, and identifies the content type of files with the parameter `DiscoverContent`. For the XML file `File.xml` its root element will be output:

```
<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 (deprecated)

`JSON Converter`: Converts a text document that contains structured data in the *JavaScript Object Notation* (JSON) into an XML document with a root element `<JSON>`, and vice versa, it converts an XML document into a UTF-8-encoded JSON document.

- This adater is deprecated: Please use the JSON Converter 2.0 instead!
- Former name until *X4 BPM Suite* version 4.5.1: *JSON Converter Adapter*

Please note when converting XML to JSON:

- Processing instructions, comments, entities, and CDATA sections in the input XML document cannot be converted.
- To create arrays in JSON out of XML element structures, add the attribute `class="array"` to the corresponding element.
- To suppress empty arrays in JSON, add an attribute `class="object"` to the corresponding XML element, e. g. `<element class="object"/>` will be `element:null` in JSON.
- For more information to the used library, see http://json-lib.sourceforge.net/usage.html.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** • *JSON2XML*: Convert a JSON document into XML • *XML2JSON*: Convert an XML document into JSON

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *`de.softproject.integration.adapter.json.JSONConverter`*: Main class (Default)
classicMode	Defines that the same XML and JSON structures are used as in previous X4 releases. **Possible values:** `true`: Classic mode is used `false`: Classic mode is not used (Default)

10.24.7 ZIP Compression (Deprecated)

`ZIP Compression`: Creates a ZIP archive with one or multiple files from data deposited in an adapter-specific XML structure, and saves the created ZIP archive into a local folder.

> ⚠ The *ZIP Compression* adapter is discontinued with version 6.2.0 of the X4 BPMS and will not be supported in future versions of the X4 BPMS. Please use the Zip Converter instead.

Properties

Operation	Defines the operation executed by the function adapter **Possible values:** *`Zip`*: Create ZIP archive from data of the input XML document

Parameter

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.zipadapter.ZipAdapter*: Main class (Default)

directory	Destination folder for the created ZIP archive ⓘ Since X4 BPMS release 5.0 it is no longer required to add a trailing slash character to separate the directory from the ZIP file name. **Possible values:** Any valid string (e.g. *%USER_COLLECTION%\Project\Folder*)
zipName	File name of the created ZIP archive **Possible values:** Any valid file name (e. g. *Output.zip*)

Input

This function adapter expects an adapter-specific input XML structure. It contains the file names and the Base64-encoded content of each file in the ZIP archive. Each `<document/>` element corresponds to a file in the new ZIP archive.

```
<data>
    <documents>
        <document>
            <filename><!--File name--></filename>
            <content><!--Base64-encoded data--></content>
        </document>
    </documents>
</data>
```

10.24.8 ZIP Folder Compression (Deprecated)

`ZIP Folder Compression`: Creates a ZIP archive from a folder, and outputs the ZIP archive as binary document with the MIME type *Application/ZIP*.

⚠ The *ZIP Folder Compression* adapter is discontinued with version 6.2.0 of the X4 BPMS and will not be supported in future versions of the X4 BPMS. Please use the Zip Converter instead.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** *Execute*: Create a ZIP archive from the selected folder

Parameter

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.zipfolderadapter.ZipFolderAdapter*: Main class (default)
zipFolderPath	Path to the folder that will be compressed into a ZIP archive **Possible values:** • Any valid string (e. g. *%USER_COLLECTION%\Project\Folder*) • Path without / or \ at the end: The created ZIP archive will contain a subfolder with the selected folder and all its contents • Path with / respectively \ at the end: The created ZIP archive will only contain the contents of the selected folder (but not the folder itself) • *(not set)*: X4 Server tries to retrieve a string from an input XML document matching the XPath expression `/X4/ZipFolderAdapter/zipfolderpath`

10.24.9 ZIP in Memory Compression (Deprecated)

`ZIP in Memory Compression`: Creates a ZIP archive with one or multiple files from data deposited in an adapter-specific XML structure, and keeps the complete data in the memory, i.e. no file system access is required.

> ⚠ The *ZIP in Memory Compression* adapter is discontinued with version 6.2.0 of the X4 BPMS and will not be supported in future versions of the X4 BPMS. Please use the Zip Converter instead.

Properties

Operation	Defines the operation executed by the adapter **Possible values:** • *Zip*: Create ZIP archive from data deposited in the input XML document in the memory • *Unzip*: Unzip the ZIP archive into the memory

Parameters

Adapter	Main adapter class (do not change!) **Possible values:** *de.softproject.integration.adapter.zipinmemory.ZipInMemory*: Main class (default)
encoding	Character encoding for file/path names **Possible values:** • Any valid encoding (e.g. *UTF-8*) • *CP850*: MS-DOS Germany (generated by MS Windows) (default) • *(empty)*: UTF-8 (for downward compatibility with previous X4 Server release versions)

Input

If property `Operation` is set to *Zip*, the adapter expects an adapter-specific input XML structure. It contains the file names and the Base64-encoded contents of all files in the ZIP archive. Each `<File>` element corresponds to a file and each `<Directory>` element to an empty folder in the ZIP archive.

```
<Archiv>
    <File name="File name" comment="Optional comment">
        <!--Base64-encoded data-->
    </File>
    <Directory name="Folder name/" />
</Archiv>
```

Output

If the property `Operation` is set to *Unzip*, the adapter creates an adapter-specific XML structure from a ZIP archive.

```
<Archiv>
    <File name="Data.txt">Q29udGVudCBvZiB0aGlzIHRleHQgZmlsZQ==</File>
    <Directory name="Folder/" />
</Archiv>
```

Explanation:

- Each element `<Directory>` corresponds to an (empty) folder in the ZIP archive.
- Each element `<File>` corresponds to a file and saves the Base64-encoded file content as value.
- The attribute `name` specifies the file or folder name.

- The attribute `size` specifies the file size, if it is deposited within the ZIP archive.