

Event Replicator for Adabas

Adabas Replication Service for webMethods Integration Server

Version 10.11

October 2022

This document applies to Event Replicator for Adabas Version 10.11 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2022 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: ARF-ARS-1011-20221108

Table of Contents

Preface	v
1 About this Documentation	1
Document Conventions	2
Online Information and Support	2
Data Protection	3
2 General Information	5
3 Installation	7
4 License	9
5 Configuration - Integration Server	11
Packages	12
Enable Autostart of EntireX Adapter	13
6 Configuration - EntireX Adapter	15
Creating and Managing Connections	16
Creating or Managing an Adabas Replication Listener Connection with the Broker as External Process	17
Creating or Managing an Adabas Replication Direct Listener Connection	20
7 Using the Adabas Replication Wizard	23
Introduction	24
Configuring a Listener	24
8 Create Document Type from Adabas File	25
9 Using the Software AG Designer	29
Generated Objects	30
Adabas Metadata Wizard	30
10 Common Scenarios	35
11 Common Usage Scenarios	37
12 Common Deployment Scenarios	39
13 Simple Deployment Using Direct RPC	41
14 Deployment Using EntireX Broker	43
15 Scenario: Notifying a User after an Adabas Database Record is Updated	45
Step 1 - Configure your Integration Server	47
Step 2 - Create Listener Objects with the Adabas Replication Wizard	49
Step 3 - Create Processing Logic for Flow Services	51
16 Scenario: Notifying a User after an Adabas Database Record is Updated	55
Step 1 - Configure your Integration Server	57
Step 2 - Create Listener Objects with the Adabas Replication Wizard	59
Step 3 - Create Processing Logic for Flow Services	61
17 Troubleshooting	65

Preface

This document provides information about the Adabas Replication Service for webMethods Integration Server (ARS). It is organized as follows:

<i>General Information</i>	Contains general information about the Adabas Replication Service for webMethods Integration Server and how it works.
<i>Installation</i>	Contains installation information.
<i>License</i>	Provides licensing information.
<i>Configuration - Integration Server</i>	Describes configuration required in the webMethods Integration Server.
<i>Configuration - EntireX Adapter</i>	Describes configuration required in the webMethods EntireX Adapter and how to establish and manage connections.
<i>Using the Adabas Replication Wizard</i>	Describes how to establish and manage listener definitions using the Adabas Replication Wizard.
<i>Create Document Type from Adabas File</i>	Describes how to create document types from an Adabas file.
<i>Using the Software AG Designer</i>	Describes how to use the Software AG Designer to view and edit objects generated with the Adabas Replication Wizard.
<i>Common Usage Scenarios</i>	Common usage scenarios with the Adabas Replication Service.
<i>Common Deployment Scenarios</i>	Common deployment scenarios with the Adabas Replication Service.

1 About this Documentation

- Document Conventions 2
- Online Information and Support 2
- Data Protection 3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Monospace font	Identifies service names and locations in the format <code>folder.subfolder.service</code> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

In addition, you can also access the cloud product documentation via <https://www.software-ag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at <https://knowledge.softwareag.com>.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at <https://tech-community.softwareag.com>. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/publishers/softwareag> and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

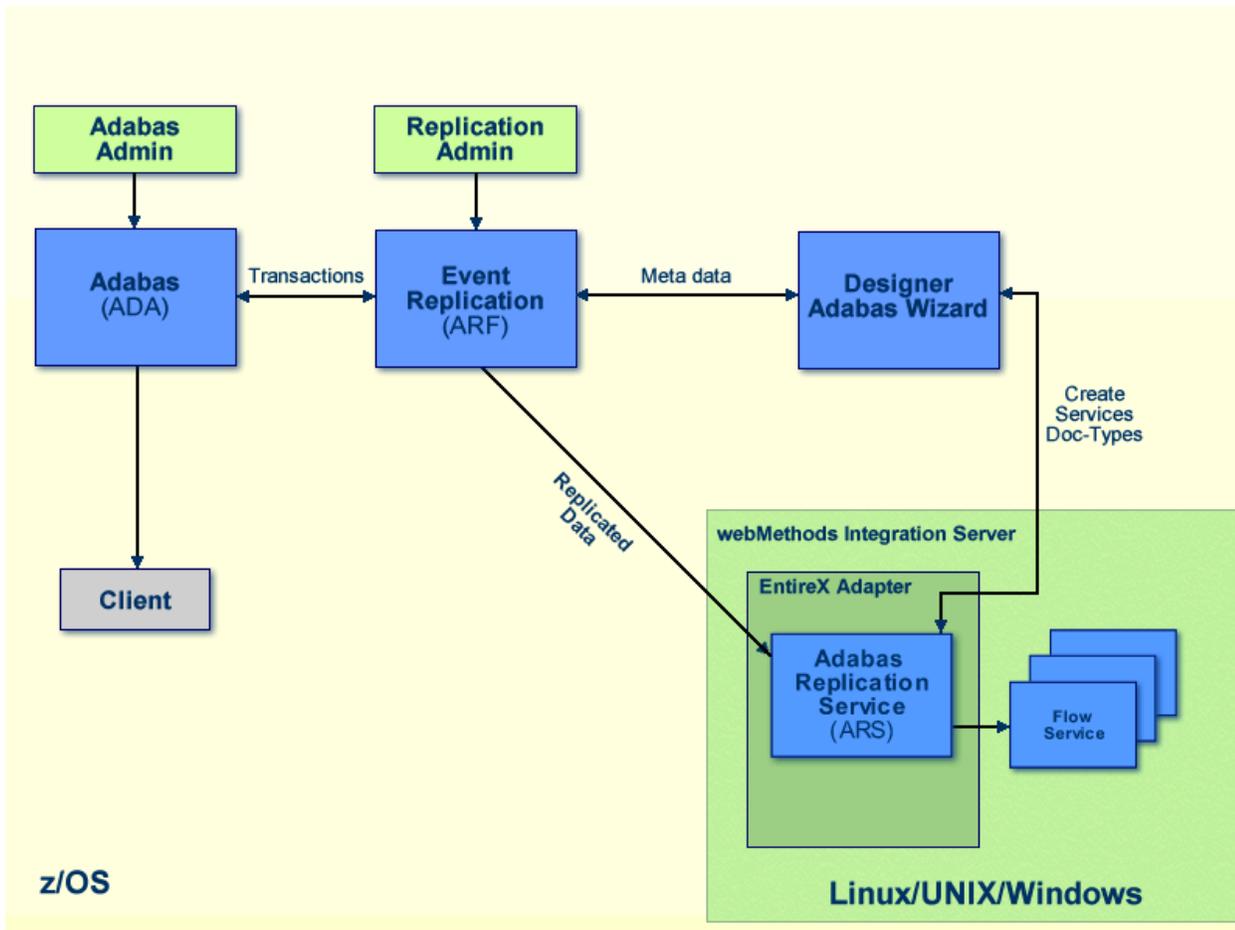
2 General Information

The purpose of sending changed data to the webMethods Integration Server is to be able to process data and store it in the final destination. Very often, only a subset of the changed data is required in the target system, or the changes are used as triggers for reading and processing data that is taken from the database - based on the changes that happened.

With this release an optimized data replication from Adabas on mainframe to webMethods Integration Server is available. The solution is called Adabas Replication Service for webMethods Integration Server (ARS) and is based on the EntireX Adapter running in the context of the Integration Server but bypassing the Event Replicator Target Adapter (ART).

The EntireX Adapter receives the changed data and converts it to the iData format used in the Integration Server context. The converted data is available to a service which can be enhanced to process the data to the required needs.

From this point on, the licensed Integration Server functionality is available and can be used to process the data. A new user interface is available to read the metadata from the Event Replicator Server on the mainframe and create document types to be used in the services.



This documentation only describes the Replication functionality for the EntireX Adapter. For details concerning the EntireX Adapter itself, refer to the documentation of this product.

About webMethods Integration Server

webMethods Integration Server is a run-time server that provides an environment for the orderly, efficient, and secure, execution of services. It supports the integration of diverse services and facilitates communication between systems. Integration Server decodes client requests, identifies the requested services, invokes the services, passes data to them in the expected format, encodes the output produced by the services, and returns output to the clients. Additionally, Integration Server authenticates clients, verifies that they are authorized to execute the requested service, maintains audit-trail logs, and promotes throughput using facilities such as service results caching.

3 Installation

To use the Adabas Replication Service for webMethods Integration Server, the following products must be installed:

- webMethods Integration Server 10.3 or above.



Note: webMethods Integration Server is not part of the Adabas Replication Service for webMethods Integration Server package and requires a separate licensing.

- webMethods EntireX Adapter 10.3 or above.

To develop your own Integration Service services using the Software AG Designer the following products/components must be installed along with the Software AG Designer:

- Adabas Event Replicator 10.3 or above.
- Service Development 10.3 or above.

4 License

To use the Adabas Replication Service for webMethods Integration Server a separate license is required. The license file can be maintained in the **License Information** node of the EntireX Adapter.

5 Configuration - Integration Server

- Packages 12
- Enable Autostart of EntireX Adapter 13

This section provides information on the basic configuration steps required on the webMethods Integration Server in order to use the Adabas Replication Service for webMethods Integration Server.

Packages

Before you begin, ensure that the EntireX Adapter package is available on the webMethods Integration Server.

➤ **To ensure that the EntireX Adapter package is available on the webMethods Integration Server**

1 In the webMethods Integration Server Administration interface go to **Packages > Management**.

The screenshot shows the 'WEBMETHODS Integration Server' administration interface. On the left is a navigation menu with the following items: Server (Scheduler, Service Usage, Statistics), Packages (Management, Publishing, Subscribing), Solutions, Adapters, webMethods Cloud, Security, and Settings. The 'Management' option under 'Packages' is selected. The main content area is titled 'Packages > Management' and contains a list of links: Install Inbound Releases, Activate Inactive Packages, Recover Packages, Browse Folders, View Locked Elements, and Filter Packages. Below this is a 'Package List' table.

Package Name	Home	Reload	Enabled	Loaded
Default			✓ Yes	✓ Yes
Event-Replication			✓ Yes	✓ Yes
WmART			✓ Yes	✓ Yes
WmARTextDC			✓ Yes	✓ Yes
WmAssetPublisher			✓ Yes	✓ Yes
WmCloud			✓ Yes	✓ Yes
WmEntireX			✓ Yes	✓ Yes
WmFlatFile			✓ Yes	✓ Yes

2 Click on **WmEntireX** for details. A screen similar to the following should be displayed:

WEBMETHODS
 Integration Server

- ▼ **Server**
 - Scheduler
 - Service Usage
 - Statistics
- ▶ **Logs**
- ▼ **Packages**
 - Management**
 - Publishing
 - Subscribing
- ▶ **Solutions**
- ▼ **Adapters**
 - EntireX Adapter ...
- ▶ **webMethods Cloud**
- ▶ **Security**
- ▶ **Settings**

Packages > Management > WmEntireX

- [Return to Packages Management](#)
- [Browse services in WmEntireX](#)

Package Information	
Package Name	WmEntireX
Version	10.3.0.0.640
Build	640
Minimum Version of JVM	1.8
Package List ACL	Default
Patches Included	
Description	EntireX Adapter
Publisher	Software AG
Created on	28 Aug 2018 17:20:07
Elements Loaded	106
Elements Not Loaded	0
Startup Services	com.softwareag.entirex.wmadapter.Admin:startUp
Shutdown Services	com.softwareag.entirex.wmadapter.Admin:shutDown
Replication Services	None
Packages on which this package depends	WmART;*,* , WmPublic;10.1
Packages that depend on this package	None
Subscribers	None

Enable Autostart of EntireX Adapter



Note: This is only required if you want to use Direct RPC Administration.

It is recommended to enable the autostart option of the EntireX Adapter in the webMethods Integration Server.

➤ To enable the EntireX autostart option

- 1 In the webMethods Integration Server Administration interface go to **EntireX Adapter > Direct RPC Administration**.

WEBMETHODS
 Integration Server

EntireX Adapter
[Connections](#)
[Listeners](#)
[Listener Notifications](#)
[Adapter Settings](#)
[Direct RPC Administration](#)
[Application Monitoring](#)
[Connections Information](#)
[Services Information](#)
[Listeners Information](#)
[License Information](#)
[Adabas Replication Wizard](#)
[Create Document Type from Adabas File](#)
[About](#)

Adapters > EntireX Adapter > Direct RPC Administration

[Refresh](#)

Configuration

Status	Running	Stop
Logging Level	Error	Change
TCP Port Number	1971 listening	
SSL Port Number		
Keystore Alias (for SSL)		
Auto Start	Enabled	Disable

Default Encoding for RPC Clients and Servers

Windows and Unix	ISO-8859-1	Change
z/OS and z/VSE	CP037	Change
BS2000	CP273	Change

Monitoring

Name	Active	Maximum	Total
Servers	0	0	0
RPC Requests	0	0	0
Socket Connections	0	0	0
Secure Socket Connections	0	0	0
Direct Connections	0	0	0
Calls	-	-	0
Conversations	0	0	0

Services

Name	Type	Instances	Requests	Wait for Server	Action
REPTOR/SERVER/SERVICE	Adabas Replication Listener	1	0	0 (0%)	

Servers

Service Name	Type	Name	Host	Version	Start Time	Action
--------------	------	------	------	---------	------------	--------

- In the **Configuration** menu go to the **Auto Start** option and check whether it is set to **Enabled**. If it is not yet enabled, click on **Enable**.

If set to **Enabled**, the EntireX Adapter is launched automatically when the webMethods Integration Server starts up.

6 Configuration - EntireX Adapter

- Creating and Managing Connections 16
- Creating or Managing an Adabas Replication Listener Connection with the Broker as External Process 17
- Creating or Managing an Adabas Replication Direct Listener Connection 20

This section provides information on the configuration steps required in the EntireX Adapter in order to use the Adabas Replication Service for webMethods Integration Server.

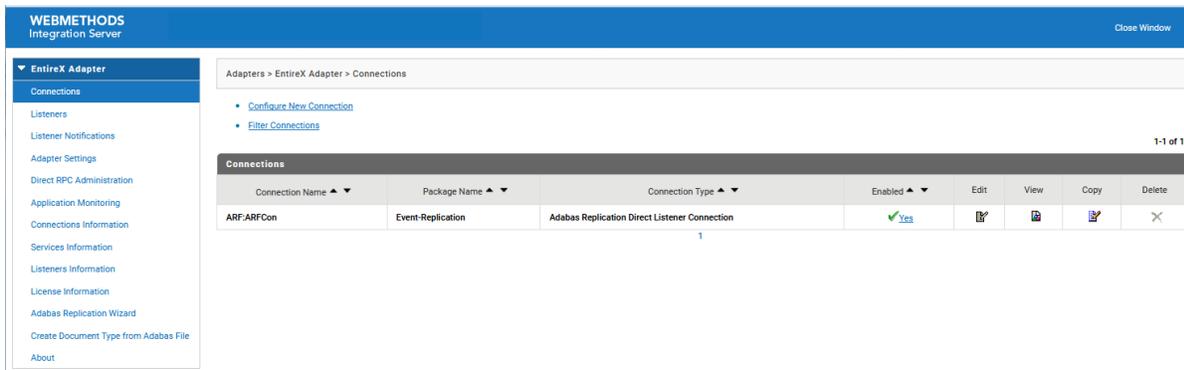
Creating and Managing Connections

In the EntireX Adapter menu you can manage existing connections and create new connections. The steps required to establish a new connection and the settings that can be made depend on the connection type.

The following connection-specific instructions apply when setting up a new or managing an existing connection unless mentioned otherwise.

> To create a new connection or manage an existing connection

- 1 In the **EntireX Adapter** menu go to **Connections**.



- 2 Click on **Configure New Connection** to create a new connection

Or:

Choose an existing connection and click on **Edit**.

For a new connection, the following screen is shown:

The screenshot shows the WEBMETHODS Integration Server interface. On the left is a navigation menu with 'EntireX Adapter' expanded to show 'Connections'. The main area displays 'Adapters > EntireX Adapter > Connection Types' with a link to 'Return to EntireX Adapter Connections'. Below is a table of connection types:

Connection Type	Description
EntireX RPC Connection	Connection to an RPC Server using the EntireX Broker
EntireX Direct RPC Connection	Direct Connection to an RPC Server
EntireX Reliable RPC Connection	Connection to an Asynchronous RPC Server using the EntireX Broker
EntireX RPC Listener Connection	Connection to an RPC Client using the EntireX Broker
EntireX Direct RPC Listener Connection	Direct Connection to an RPC Client
EntireX Reliable RPC Listener Connection	Direct connection to an Asynchronous RPC Client
EntireX Direct Reliable RPC Listener Connection	Connection to an Asynchronous RPC Client using Direct RPC
IMS Connect Connection	Connection to IMS Connect
CICS ECI Connection	Connection to CICS ECI
CICS Socket Listener Connection	Connection to CICS Socket Listener
Applinx Connection	Connection to Applinx Server
ACI Server Connection	Connection to ACI Server
COBOL Converter Connection	COBOL Converter Connection
Adabas Replication Listener Connection	Connection to an Adabas Replication Server using the EntireX Broker
Adabas Replication Direct Listener Connection	Direct Connection to an Adabas Replication Server

- 3 Select the type of connection you want to establish.
- 4 For an existing connection or when the type of connection has been selected for a new connection, the next screen to be shown depends on the type of connection. Please refer to the following sections for details.

Creating or Managing an Adabas Replication Listener Connection with the Broker as External Process

➤ To create an Adabas Replication Listener Connection

- 1 In the **EntireX Adapter** menu go to **Connections**.

Click on **Adabas Replication Listener Connection**.

The following screen is shown:

Configure Connection Type > EntireX Adapter	
Package	Default <input type="button" value="v"/>
Folder Name	<input type="text"/>
Connection Name	<input type="text"/>
Connection Properties	
Broker ID	<input type="text"/>
Server Address	REPTOR/SERVER/SERVICE
Logon User	<input type="text"/>
Logon Password	<input type="text"/>
Retype Logon Password	<input type="text"/>
Single Conversation Mode	true <input type="button" value="v"/>
Connection Management Properties	
Enable Connection Pooling	true <input type="button" value="v"/>
Minimum Pool Size	1
Maximum Pool Size	10
Pool Increment Size	1
Block Timeout (msec)	1000
Expire Timeout (msec)	1000
Startup Retry Count	0
Startup Backoff Timeout (sec)	10
<input type="button" value="Save Connection"/>	

2 Provide the required parameter values. Parameters and their possible values are described in the following table:

Parameter	Description
Package	Name of the package where the connection is stored.
Folder Name	Name of the folder where the connection is stored.
Connection Name	The name of the connection.
Broker ID	The ID of the broker you want to connect to. This ID consists of a host and an optional port. Default for the port is 1971.
Server Address	The address of the RPC server registered to the broker above. The address is given in the format <class>/<server>/<service>.
Logon User	The name of the user to log on to the broker.
Logon Password	The password for the user above.
Retype Logon Password	Verification of the password.
Single Conversation Mode	Enables or disables the use of Single Conversation Mode.

Parameter	Description
	Default: true.
Enable Connection Pooling	Enables the adapter to use connection pooling. Default: true. Refer to <i>Connection Pooling</i> under <i>EntireX Adapter Connections</i> in the <i>webMethods EntireX</i> documentation for more information on connection pooling.
Minimum Pool Size	If connection pooling is enabled, this field specifies the minimum number of connection objects that remain in the connection pool at all times. When the adapter creates the pool, it creates this number of connections. Default: 1.
Maximum Pool Size	The maximum number of connection objects that can exist in the connection pool. The adapter will reuse any inactive connections in the pool or, if all connections are active and the connection pool has reached its maximum size, the adapter will wait for a connection to become available. Default: 10.
Pool Increment Size	If connection pooling is enabled, this field specifies the number of connections by which the pool will be incremented if connections are needed, up to the maximum pool size. Default: 1.
Block Timeout	If connection pooling is enabled, this field specifies the number of milliseconds that the Integration Server will wait to obtain a connection before it times out and returns an error. Default: 1000.
Expire Timeout	If connection pooling is enabled, this field specifies the number of milliseconds that an inactive connection can remain in the pool before it is closed and removed from the pool. For example, to specify 10 seconds, specify 10000. Enter 0 to specify no timeout. Default: 1000. Note: The adapter will never violate the Minimum Pool Size parameter. These connections remain in the pool regardless of how long they are inactive.
Startup Retry Count	The number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails. Default: 0.
Startup Backoff Timeout	The number of seconds that the system should wait between attempts to initialize the connection pool.

- 3 Click on **Save Connection**.

Creating or Managing an Adabas Replication Direct Listener Connection

➤ To create an Adabas Replication Direct Listener Connection

- 1 In the **EntireX Adapter** menu go to **Connections**.

Click on **Adabas Replication Direct Listener Connection**.

The following screen is shown:

Configure Connection Type > EntireX Adapter	
Package	Default ▾
Folder Name	<input type="text"/>
Connection Name	<input type="text"/>
Connection Properties	
Server Address	REPTOR/SERVER/SERVICE
Connection Management Properties	
Enable Connection Pooling	true ▾
Minimum Pool Size	1
Maximum Pool Size	10
Pool Increment Size	1
Block Timeout (msec)	1000
Expire Timeout (msec)	1000
Startup Retry Count	0
Startup Backoff Timeout (sec)	10
<input type="button" value="Save Connection"/>	

- 2 Provide the required parameter values. Parameters and their possible values are described in the following table:

Parameter	Description
Package	Name of the package where the connection is stored.
Folder Name	Name of the folder where the connection is stored.
Connection Name	The name of the connection.
Server Address	The address of the RPC server. The address is given in the format <class>/<server>/<service>.

Parameter	Description
Enable Connection Pooling	<p>Enables the adapter to use connection pooling.</p> <p>Default: true.</p> <p>Refer to <i>Connection Pooling</i> under <i>EntireX Adapter Connections</i> in the <i>webMethods EntireX</i> documentation for more information on connection pooling.</p>
Minimum Pool Size	<p>If connection pooling is enabled, this field specifies the minimum number of connection objects that remain in the connection pool at all times. When the adapter creates the pool, it creates this number of connections.</p> <p>Default: 1.</p>
Maximum Pool Size	<p>The maximum number of connection objects that can exist in the connection pool. The adapter will reuse any inactive connections in the pool or, if all connections are active and the connection pool has reached its maximum size, the adapter will wait for a connection to become available.</p> <p>Default: 10.</p>
Pool Increment Size	<p>If connection pooling is enabled, this field specifies the number of connections by which the pool will be incremented if connections are needed, up to the maximum pool size.</p> <p>Default: 1.</p>
Block Timeout	<p>If connection pooling is enabled, this field specifies the number of milliseconds that the Integration Server will wait to obtain a connection before it times out and returns an error.</p> <p>Default: 1000.</p>
Expire Timeout	<p>If connection pooling is enabled, this field specifies the number of milliseconds that an inactive connection can remain in the pool before it is closed and removed from the pool.</p> <p>For example, to specify 10 seconds, specify 10000. Enter 0 to specify no timeout.</p> <p>Default: 1000.</p> <p>Note: The adapter will never violate the Minimum Pool Size parameter. These connections remain in the pool regardless of how long they are inactive.</p>
Startup Retry Count	<p>The number of times that the system should attempt to initialize the connection pool at startup if the initial attempt fails.</p> <p>Default: 0.</p>
Startup Backoff Timeout	<p>The number of seconds that the system should wait between attempts to initialize the connection pool.</p>

3 Click on **Save Connection**.

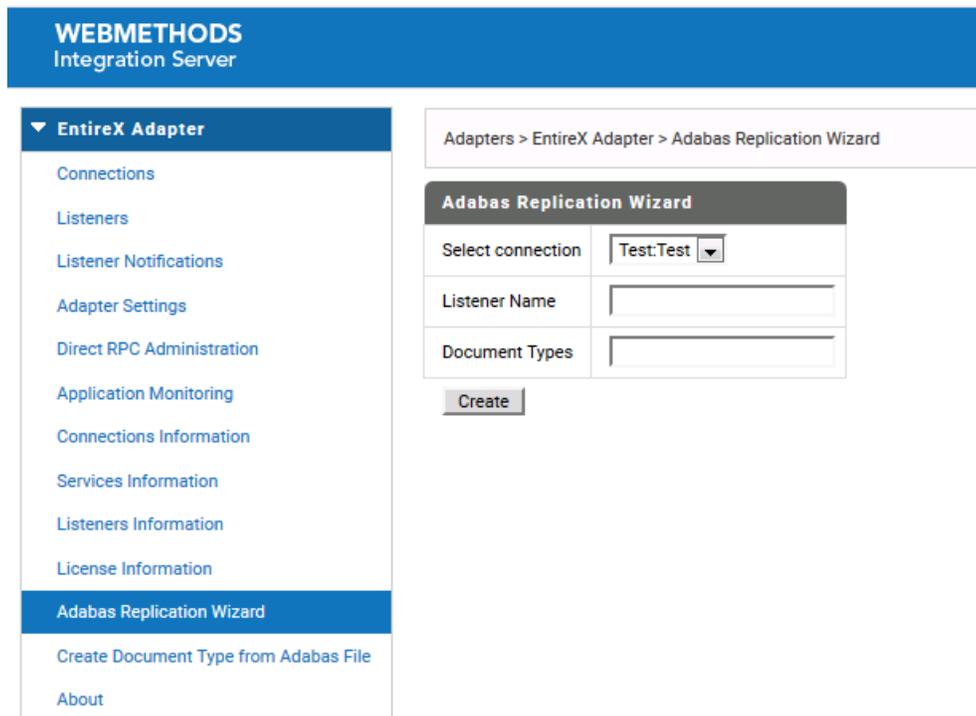
Once you have established a connection, you have to define a listener for each connection you want to use with the Adabas Replication Service for webMethods Integration Server. Defining listeners is described in the following section *Using the Adabas Replication Wizard*.

7 Using the Adabas Replication Wizard

- Introduction 24
- Configuring a Listener 24

Introduction

The Adabas Replication Wizard can be used to generate a listener and all required objects for an existing connection. The listener will be stored in the same location as the connection.



Configuring a Listener

Parameter	Description
Select connection	Select the connection for which the objects are generated. Note: The connection must be enabled so that it can be selected from the drop down menu.
Listener Name	Name of the listener.
Document Types	The wizard is able to generate listener services from existing Document types. These Document type must be entered in the Integration Server name schema. More than one Document type must be separated by a comma.

8

Create Document Type from Adabas File

Adabas Replication Service for webMethods Integration Server is able to generate document type and Metadata from the definitions that are stored on the Replication System file.

WEBMETHODS
 Integration Server
 default :: daeirnd42335.eur.ad.sag :: Administrator [Close Window](#)

▼ EntireX Adapter

- Connections
- Listeners
- Listener Notifications
- Adapter Settings
- Direct RPC Administration
- Application Monitoring
- Connections Information
- Services Information
- Listeners Information
- License Information
- Adabas Replication Wizard
- Create Document Type from Adabas File
- About

Adapters > EntireX Adapter > Create Document Type / Metadata

Global Format Buffer

Name	<input type="text"/>
------	----------------------

Source

Replication System File

Node	<input type="text"/>
Port	<input type="text"/>
Database Number	<input type="text"/>
File Number	<input type="text"/>

SYSRPTR Unload File

File Name	<input type="text"/>
-----------	----------------------

Target

Generate Document Type

Package	<input type="text"/>
Folder	<input type="text"/>
Generate Document Type	<input type="checkbox"/> false ▼

Generate Metadata

Connection	<input type="text"/>
Table Name	<input type="text"/>
Generate Metadata	<input type="checkbox"/> false ▼

With this function two different object types can be created from a Global Format Buffer: Document Types and Metadata.

A Document Type contains a set of fields used to define the structure and type of data in a document (IData object). For more information about the Document Type see the *webmethods Integration Server* documentation. Metadata is needed to decode the replication data from the Event Replicator for Adabas to the iData object.

The Metadata is provided when the Initial State is done. If the Metadata is not available, it can be created with this function.

Parameter	Description
Name	The name of the Global Format Buffer on the Replication System File.
Node	The network node where the Replicator system file can be found. This is the TCP/IP node or IP address where the Event Replicator Server nucleus resides.
Port	The port number where the Replicator system file can be found. This is the TCP/IP port number used by the Entire Net-Work TCP/IP Option on the mainframe.
Database Number	Database number of the Replication System File.
File Number	File number of the Replication System File.
File Name	Name of the SYSRPTR Unload File.
Package	Package in which the Document type is stored.
Folder	Folder in which the Document type is stored.
Generate Document Type	Check this box to generate Document type.
Connection	Connection name in the Integration Server naming schema.
Table Name	Optional. Only required when the table name is different from Global Format Buffer Name.
Generate Metadata	Check this box to generate Metadata.



Note: If the Replication System File is not accessible via the Entire Net-Work, the data can be generated from SYSRPTR Unload File. In this case, leave the Replication System File empty and enter the file name in **SYSRPTR Unload File**.

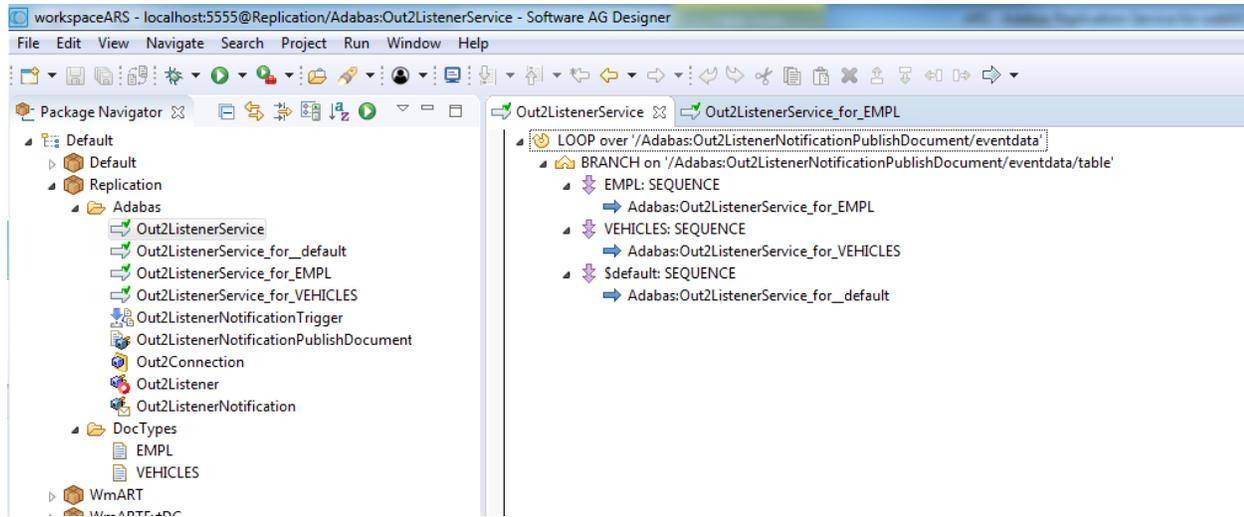
9 Using the Software AG Designer

■ Generated Objects	30
■ Adabas Metadata Wizard	30

You can use the Software AG Designer to view and edit the objects generated with the Adabas Replication Wizard.

Generated Objects

This picture shows object that were created by the Adabas Replication Wizard:



The Flow Services are the place where users should insert their logic.

Adabas Metadata Wizard

The Adabas Metadata Wizard is invoked using the **New** command.

➤ **To invoke the Adabas Metadata Wizard via the New command:**

- 1 In the top menu of the Software AG Designer click on **File > New > Other...**

Or:

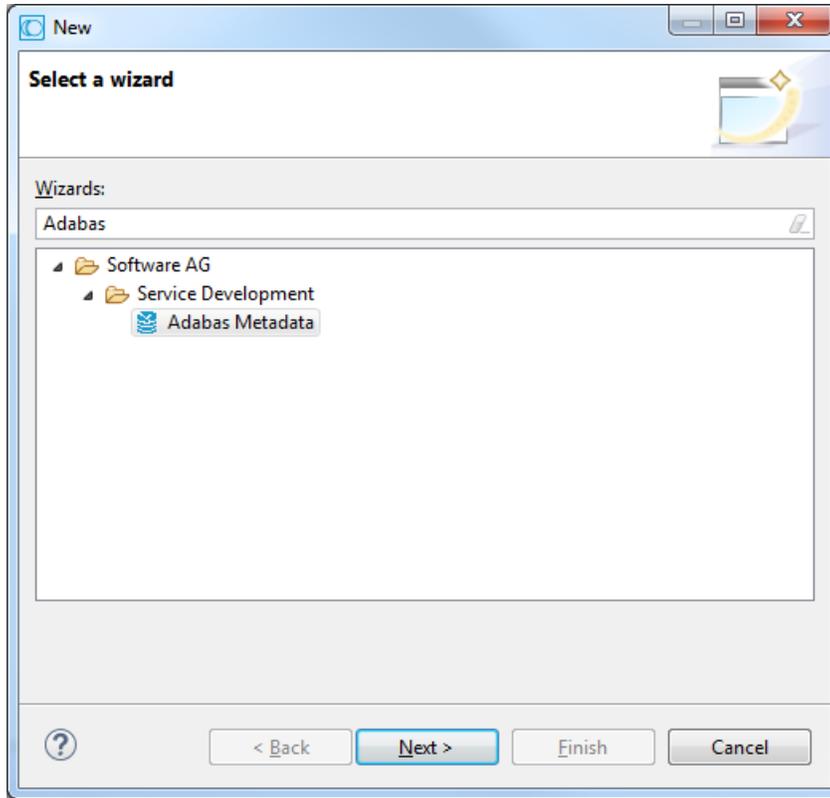
In the toolbar click on  and then select **Other...** from the menu.

Or:

Use the shortcut **CTRL+N**.

The **New** wizard is opened.

- 2 Navigate to **Software AG > Service Development** and click on **Adabas Metadata**.



The Designer Wizard is able to create Document types and Metadata.

You can either create them using a system file or read the metadata from a SYSRPTR unload file:

Source

Define the Replication Source that contains the Event Replicator definitions.

System File

Replication System File

Node:

Port:

Database Number:

File Number:

Unload File

Replication Unload File

File Name:

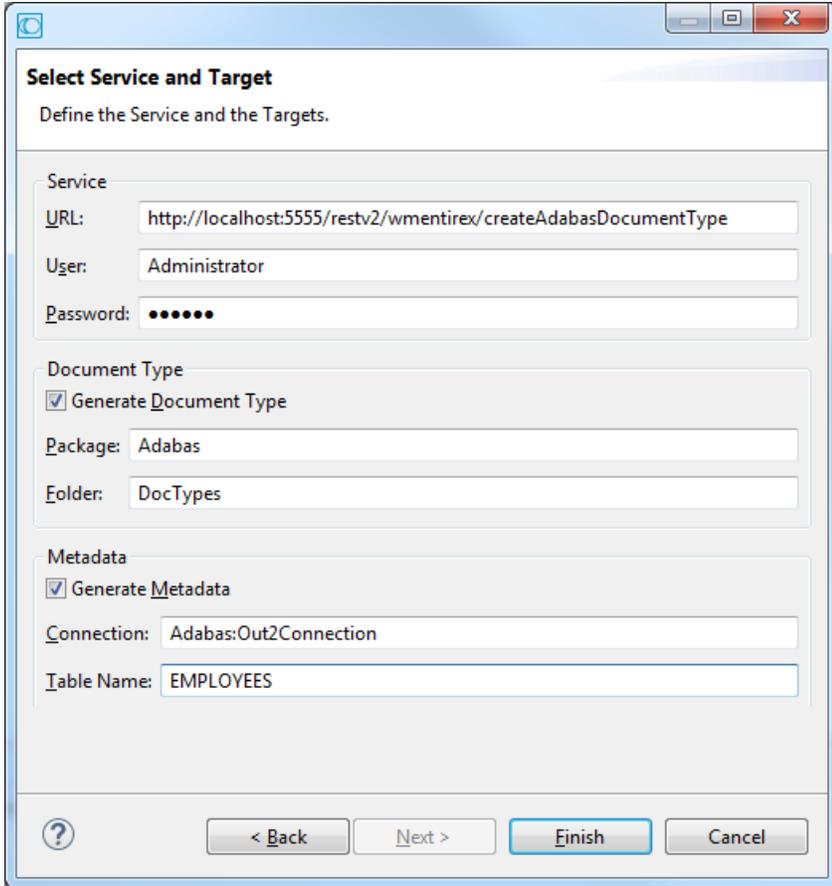
Select Format Buffer

Define the Replication Service that contains the Event Replicator definitions.

Format Buffer:

Type	Level	Field ID	Format
	1	PERSONNEL_ID	A8.0
GR	1	FULLNAME	
	2	FIRST_NAME	A20.0
	2	MIDDLE_NAME	A20.0
	2	NAME	A20.0
	1	MARSTAT	A1.0
	1	SEX	A1.0
GR	1	FULL_ADDRESS	
MU	2	ADDRESS_LINE	A20.0
	2	CITY	A20.0
	2	POSTCODE	A10.0
	2	COUNTRY	A3.0
GR	1	TELEPHONE	
	2	AREACODE	A6.0





The parameters that can be set edited are the same as described in the table of the section [Create Document Type from Adabas File](#).

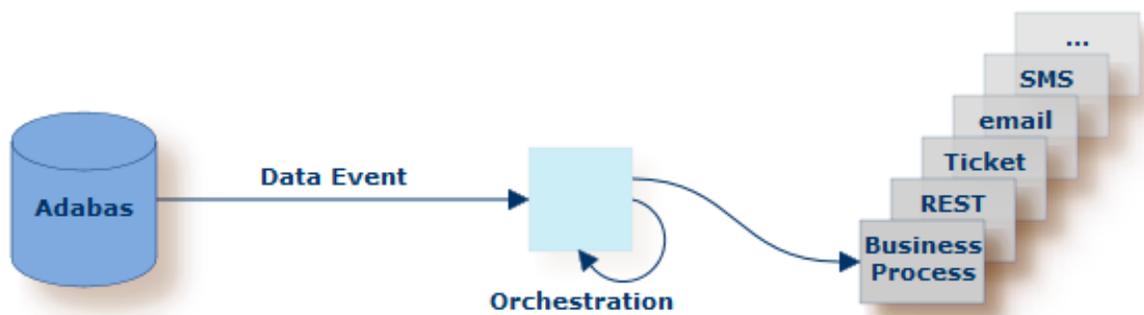
10

Common Scenarios

- For Event-based replication, for example sending an e-mail or initiating a REST call, see [Common Usage Scenarios](#).
- For all other replication scenarios, see Getting Started.

11 Common Usage Scenarios

This chapter describes common usage scenarios with the Adabas Replication Service for webMethods Integration Server.

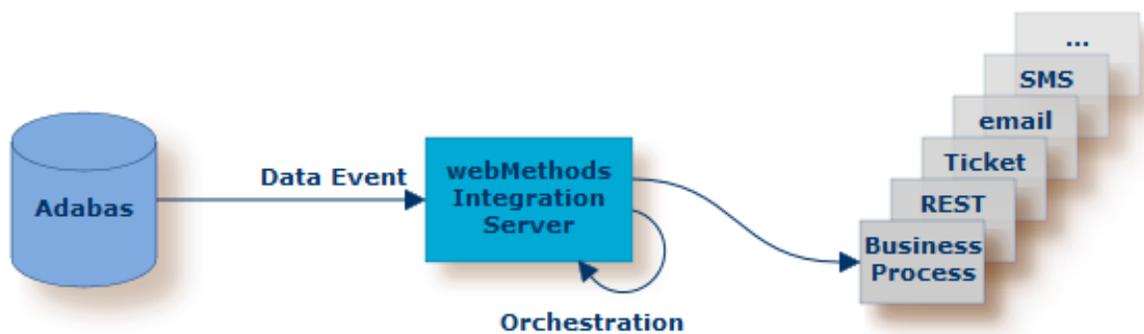


- I want to orchestrate the incoming event with other data
- I want to trigger a business process
- I want to call a REST API / resource
- I want to open a ticket in a ticketing system
- I want to send an SMS or an e-mail

To implement one of these scenarios, see [Common Deployment Scenarios](#).

12 Common Deployment Scenarios

All **Common Usage Scenarios** described here are realized via the webMethods Integration Server.

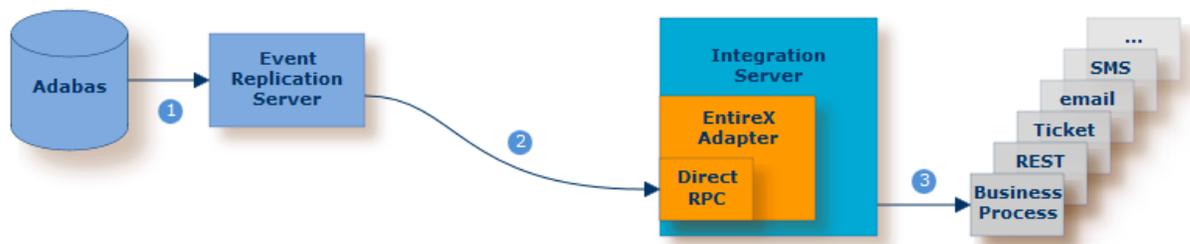


You have two architectural choices for deploying the components:

- **Simple Deployment Using Direct RPC**
- **Deployment Using EntireX Broker**

13 Simple Deployment Using Direct RPC

All **Common Usage Scenarios** are realized via the webMethods Integration Server.

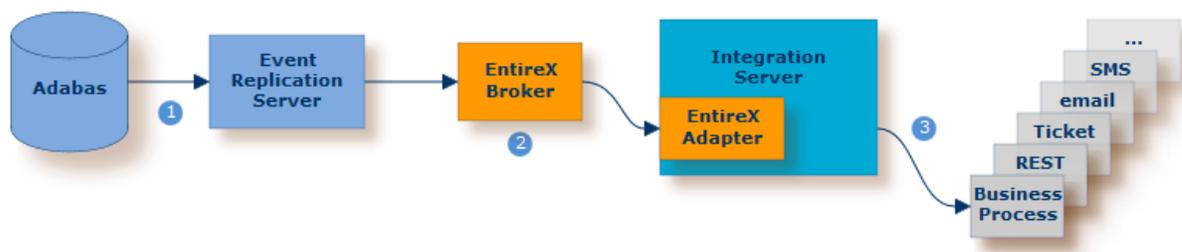


- 1 Event Replicator for Adabas listens for specific actions on the database.
- 2 Event Replicator for Adabas sends an XML message to Integration Server via Direct RPC of Integration Server
- 3 The EntireX Adapter listens to the respective message queue and triggers an IS service, for example starting a business process or notification by email.

See [Scenario: Notifying a User after an Adabas Database Record is Updated](#) for the steps required.

14 Deployment Using EntireX Broker

All **Common Usage Scenarios** are realized via the webMethods Integration Server.



- 1 Event Replicator for Adabas listens for specific actions on the database.
- 2 Event Replicator for Adabas sends an XML message to Integration Server via EntireX Broker
- 3 The EntireX Adapter listens to the respective message queue and triggers an IS service, for example starting a business process or notification by email.

See **Scenario: Notifying a User after an Adabas Database Record is Updated** for the steps required.

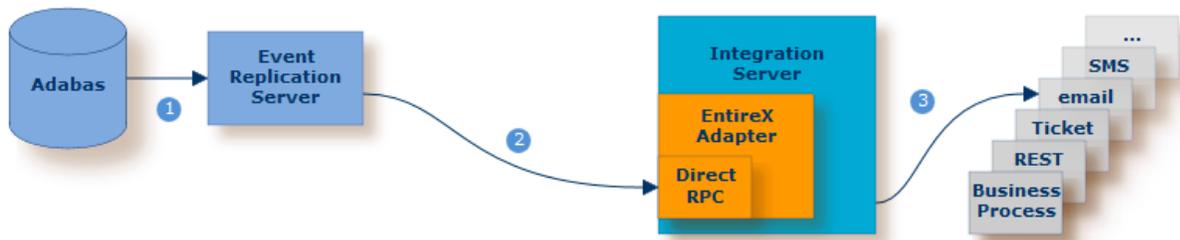
15 Scenario: Notifying a User after an Adabas Database

Record is Updated

- Step 1 - Configure your Integration Server 47
- Step 2 - Create Listener Objects with the Adabas Replication Wizard 49
- Step 3 - Create Processing Logic for Flow Services 51

Scenario: I have a mainframe Adabas database and want to notify a user when a record is updated.

Solution: Use the Event Replicator for Adabas to send a message to the webMethods Integration Server via Direct RPC, and use Integration Server logic to send an email to the target user.



- 1 Event Replicator for Adabas listens for specific actions on the database.
- 2 Event Replicator for Adabas sends an XML message to Integration Server via Direct RPC of Integration Server
- 3 The EntireX Adapter listens to the respective message queue and triggers an IS service to send the email.

This scenario requires the following Software AG products:

- Mainframe Adabas
- Event Replicator for Adabas
- webMethods Integration Server
- EntireX Adapter

Step 1 - Configure your Integration Server

- [Check the EntireX Adapter Package is Available](#)
- [Enable Autostart \(Optional\)](#)
- [Creating an Adabas Replication Direct Listener Connection](#)

Check the EntireX Adapter Package is Available

Before you begin, ensure that the EntireX Adapter package is available on the webMethods Integration Server.

➤ **To ensure that the EntireX Adapter package is available on the webMethods Integration Server**

- 1 In the webMethods Integration Server Administration interface, go to **Packages > Management**.
- 2 Click on **WmEntireX** for details. A screen similar to this should be displayed (extract):

Package Information	
Package Name	WmEntireX
Version	10.3.0.0.640
Build	640
Minimum Version of JVM	1.8
Package List ACL	Default
Patches Included	
Description	EntireX Adapter
Publisher	Software AG

Enable Autostart (Optional)

We recommend enabling the autostart option of the EntireX Adapter in the webMethods Integration Server; EntireX Direct RPC is then launched automatically when the Integration Server starts up.

> To enable the EntireX autostart option

- 1 In the webMethods Integration Server Administration interface go to **EntireX Adapter > Direct RPC Administration**.

The screenshot shows the 'EntireX Adapter' configuration interface. On the left is a navigation menu with options: Connections, Listeners, Listener Notifications, Adapter Settings, Direct RPC Administration (highlighted), Application Monitoring, and Connections Information. The main area is titled 'Configuration' and contains a table with the following data:

Property	Value	Action
Status	Running	Stop
Logging Level	Error	Change
TCP Port Number	1971 listening	
SSL Port Number		
Keystore Alias (for SSL)		
Auto Start	Enabled	Disable

- 2 In the **Configuration** menu go to the **Auto Start** option and check whether it is set to **Enabled**. If it is not yet enabled, click on **Enable**.

Creating an Adabas Replication Direct Listener Connection

> To create an Adabas Replication Direct Listener Connection

- 1 In the **EntireX Adapter** menu go to **Connections**. Click on **Adabas Replication Direct Listener Connection** to display the following screen:

The screenshot shows the 'Configure Connection Type > EntireX Adapter' screen. It features a form with the following fields:

- Package:** A dropdown menu currently set to 'Default'.
- Folder Name:** An empty text input field.
- Connection Name:** An empty text input field.
- Connection Properties:** A section header for the following field.
- Server Address:** A text input field containing the value 'REPTOR/SERVER/SERVICE'.

- 2 Add the name of the package where the connection is stored.

- 3 Add the name of the folder where the connection is stored.
- 4 Add the name of the connection.
- 5 Add the server address in the format <class>/<server>/<service>.



Note: All other parameters are optional. Normally you can use the default values for these. All parameters are described [here](#).

- 6 Click on **Save Connection**.

In the next step, create a listener for this connection, using the Adabas Replication Wizard.

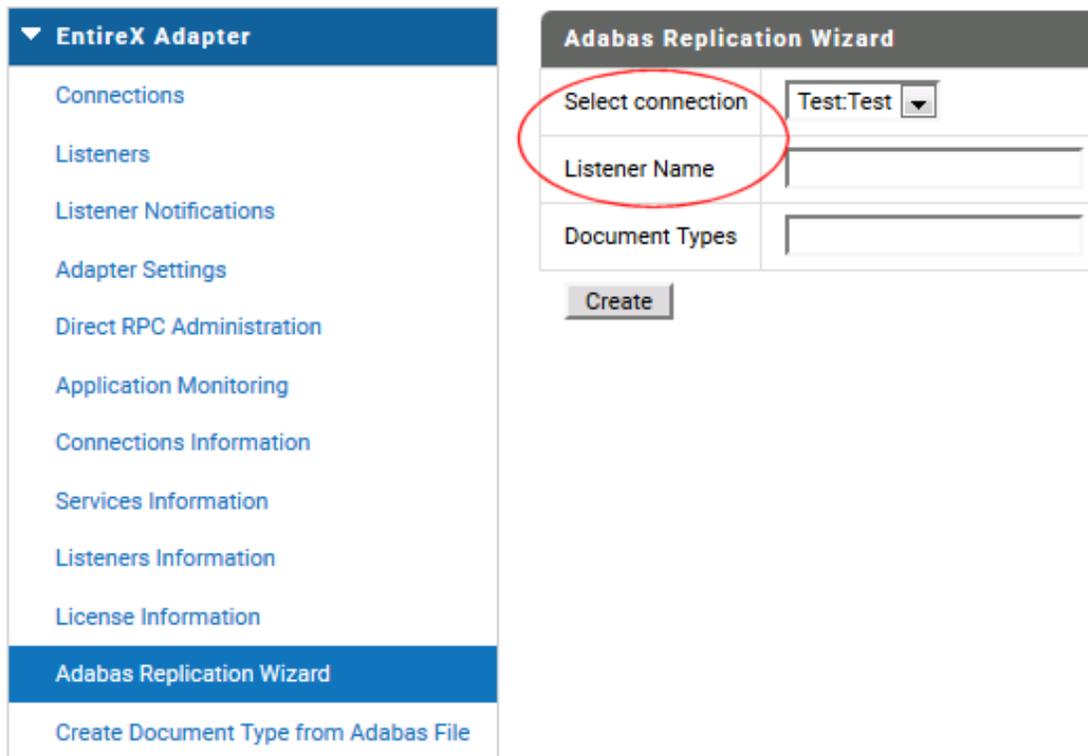
Step 2 - Create Listener Objects with the Adabas Replication Wizard

➤ To generate a listener and all required objects for an existing connection

- 1 In the webMethods Integration Server, choose **Adapters > EntireX Adapter > Connections** and make sure the connection you created in the previous step is enabled.

EntireX Adapter Listeners		
Listener Name ▲ ▼	Package Name ▲ ▼	State ▲ ▼
Group:GroupListener	Default	Disabled ▼
Out1:Out1Listener	Default	Disabled ▼
Test:TestListener	Default	Disabled ▼
Training:TrainingListener	Default	Disabled ▼
directrpc:directListener	Default	<div style="border: 1px solid red; border-radius: 50%; padding: 2px;"> Disabled ▼ Disabled Enabled </div>

- 2 In the Integration Server, choose **EntireX Adapter > Adabas Replication Wizard**.



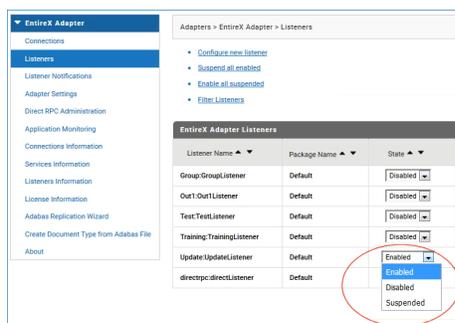
- 3 Select the connection you created previously.
- 4 Enter the name of the new listener.
- 5 Choose Create.

The listener will be stored in the same location as the connection.



Note: Document types can be optionally created to restrict the scope of the listening operation. See [Create Document Type from Adabas File](#).

- 6 Make sure the listener you created is enabled. Choose **Enabled** from the drop-down menu.

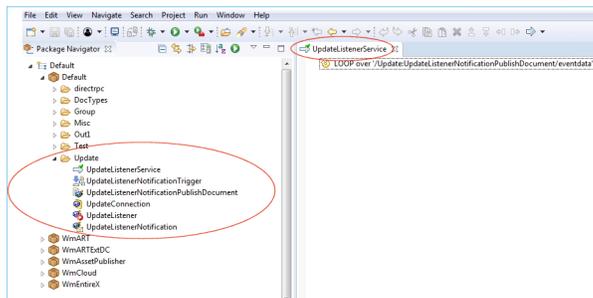


Step 3 - Create Processing Logic for Flow Services

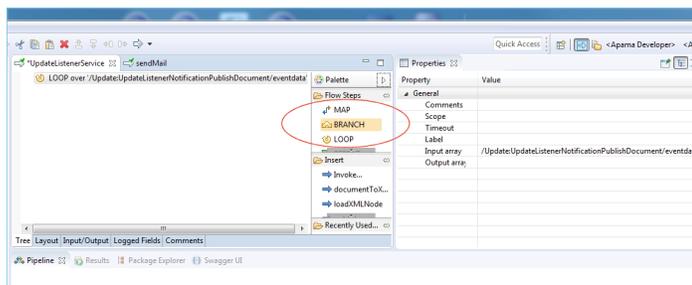
In the previous step you created the required listener objects with the Adabas Replication Wizard. Now you can add your own logic to perform the actions you require. In the example below we are sending an email with the updated data in XML format.

➤ To customize the newly created flow service

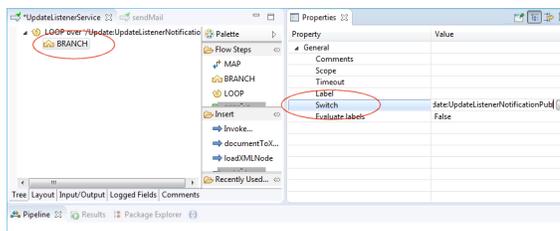
- 1 Open the flow service created in the previous step to add your custom logic.



- 2 Add a flow step **BRANCH** within the **LOOP**. This enables you to decide on a specific action for a particular field.

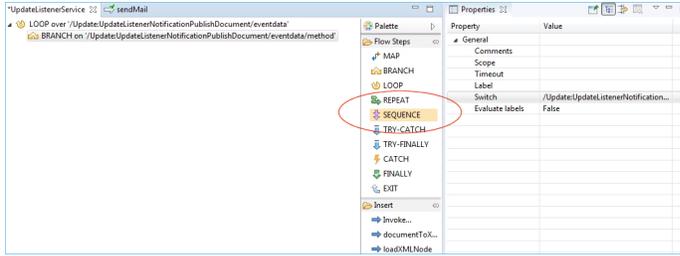


- 3 In the **Switch** property, set the path to an attribute of the event data. In this case it is the **method** attribute of the event data.

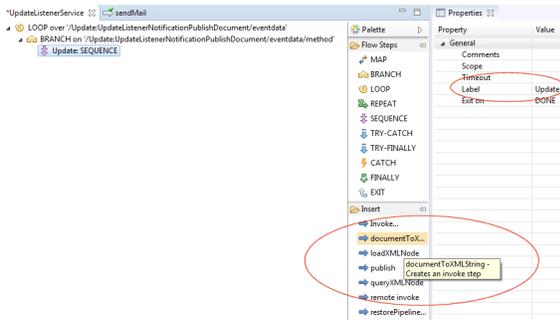


- 4 Under flow step **SEQUENCE**, specify what happens when the method has a particular value.

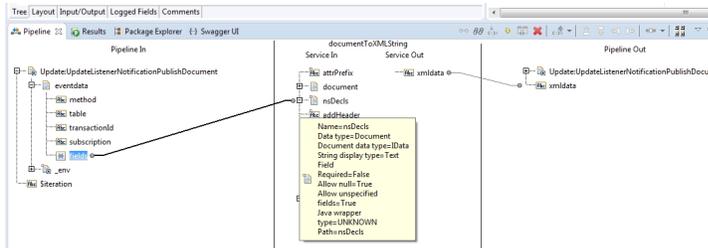
Scenario: Notifying a User after an Adabas Database Record is Updated



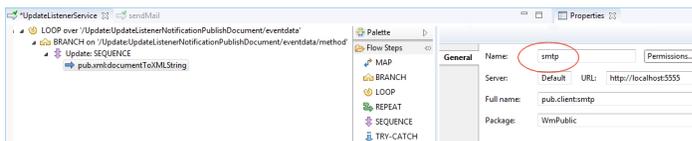
5 Set **Label** to **Update** and map document (the input format) to XML string.



6 Use the `pub.sml:documentToXMLString` service from the `WmPublic` package to document from fields an XML string that can be sent via email.



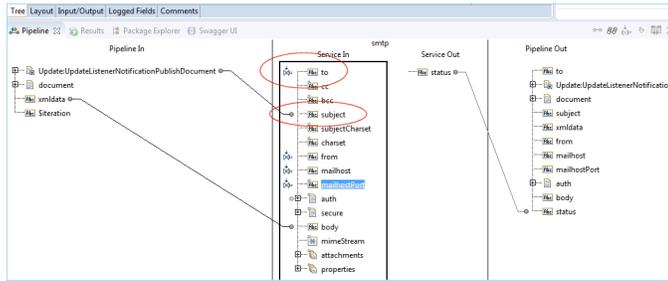
7 Specify the mail protocol, in our example "smtp". To send the mail you can use the public flow service `pub.client:smtp` from the `WmPublic` package.



8 Set the SMPT parameters.

- Some fields are coded manually, for example **to**. These fields are marked with the icon .
- Other fields are filled dynamically with values from the flow service, for example **subject**.

These fields are marked with the icon



The result of these steps is an email where

- the subject is dynamically filled with the value from the flow service
- the address is coded manually in a mapping operation
- the body is converted to XML and contains the fields that have been updated

```

EMPL
To: GER@eur.ad.sag
From: Do 01.09.2019 16:06
Cc: Gerth, Matthias

<?xml version="1.0"?>
<IDM>1128</IDM>
<CITY>Auckland</CITY>
<DEPT>SALED1</DEPT>
    
```

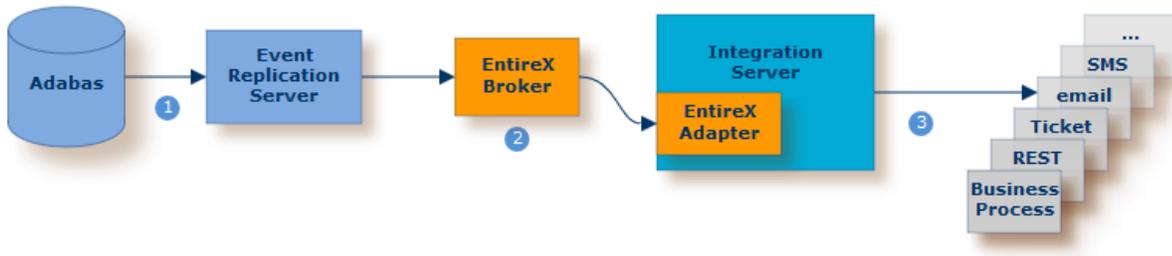

16 Scenario: Notifying a User after an Adabas Database

Record is Updated

- Step 1 - Configure your Integration Server 57
- Step 2 - Create Listener Objects with the Adabas Replication Wizard 59
- Step 3 - Create Processing Logic for Flow Services 61

Scenario: I have a mainframe Adabas database and want to notify a user when a record is updated.

Solution: Use the Event Replicator for Adabas to send a message to the webMethods Integration Server via EntireX Broker, and use Integration Server logic to send an email to the target user.



- 1 Event Replicator for Adabas listens for specific actions on the database.
- 2 Event Replicator for Adabas sends an XML message to Integration Server via EntireX Broker
- 3 The EntireX Adapter listens to the respective message queue and triggers an IS service to send the email.

This scenario requires the following Software AG products:

- Mainframe Adabas
- Event Replicator for Adabas
- webMethods Integration Server
- EntireX Broker
- EntireX Adapter

Step 1 - Configure your Integration Server

- [Check the EntireX Adapter Package is Available](#)
- [Creating an Adabas Replication Listener Connection](#)

Check the EntireX Adapter Package is Available

Before you begin, ensure that the EntireX Adapter package is available on the webMethods Integration Server.

➤ To ensure that the EntireX Adapter package is available on the webMethods Integration Server

- 1 In the webMethods Integration Server Administration interface, go to **Packages > Management**.
- 2 Click on **WmEntireX** for details. A screen similar to this should be displayed (extract):

Package Information	
Package Name	WmEntireX
Version	10.3.0.0.640
Build	640
Minimum Version of JVM	1.8
Package List ACL	Default
Patches Included	
Description	EntireX Adapter
Publisher	Software AG

Creating an Adabas Replication Listener Connection

➤ To create an Adabas Replication Listener Connection

- 1 In the **EntireX Adapter** menu go to **Connections**. Click on **Adabas Replication Listener Connection** to display the following screen:

Adapters > EntireX Adapter > Configure Connection Type

- [Return to EntireX Adapter Connection Types](#)

Configure Connection Type > EntireX Adapter

Package	Default
Folder Name	
Connection Name	

Connection Properties

Broker ID	
Server Address	REPTOR/SERVER/SERVICE
Logon User	
Logon Password	
Retype Logon Password	
Single Conversation Mode	true

- 2 Add the name of the package where the connection is stored.
- 3 Add the name of the folder where the connection is stored.
- 4 Add the name of the connection.
- 5 Add the ID of the broker you are using for this task.
- 6 Add the server address in the format <class>/<server>/<service>.
- 7 Add the user ID and password (if applicable).

 **Note:** All other parameters are optional. Normally you can use the default values for these. All parameters are described [here](#).

- 8 Click on **Save Connection**.

In the next step, create a listener for this connection, using the Adabas Replication Wizard.

Step 2 - Create Listener Objects with the Adabas Replication Wizard

➤ To generate a listener and all required objects for an existing connection

- 1 In the webMethods Integration Server, choose **Adapters > EntireX Adapter > Connections** and make sure the connection you created in the previous step is enabled.

EntireX Adapter Listeners		
Listener Name ▲ ▼	Package Name ▲ ▼	State ▲ ▼
Group:GroupListener	Default	Disabled ▼
Out1:Out1Listener	Default	Disabled ▼
Test:TestListener	Default	Disabled ▼
Training:TrainingListener	Default	Disabled ▼
directrpc:directListener	Default	Disabled ▼ Disabled Enabled

- 2 In the Integration Server, choose **EntireX Adapter > Adabas Replication Wizard**.

EntireX Adapter

- Connections
- Listeners
- Listener Notifications
- Adapter Settings
- Direct RPC Administration
- Application Monitoring
- Connections Information
- Services Information
- Listeners Information
- License Information
- Adabas Replication Wizard
- Create Document Type from Adabas File

Adabas Replication Wizard

Select connection ▼ Test:Test ▼

Listener Name

Document Types

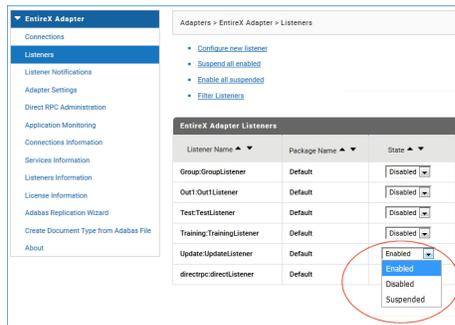
- 3 Select the connection you created previously.
- 4 Enter the name of the new listener.
- 5 Choose Create.

The listener will be stored in the same location as the connection.



Note: Document types can be optionally created to restrict the scope of the listening operation. See [Create Document Type from Adabas File](#).

- 6 Make sure the listener you created is enabled. Choose **Enabled** from the drop-down menu.

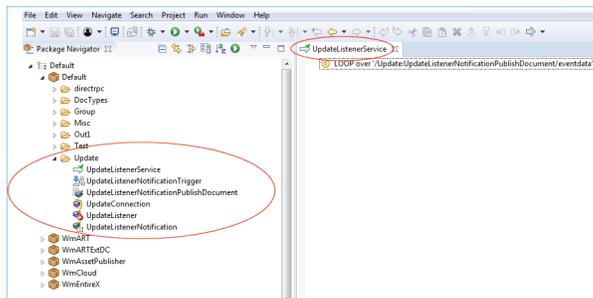


Step 3 - Create Processing Logic for Flow Services

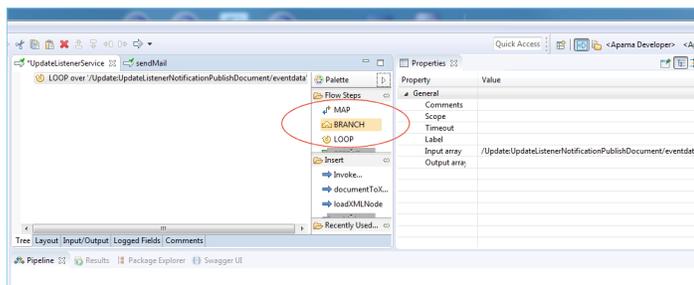
In the previous step you created the required listener objects with the Adabas Replication Wizard. Now you can add your own logic to perform the actions you require. In the example below we are sending an email with the updated data in XML format.

➤ To customize the newly created flow service

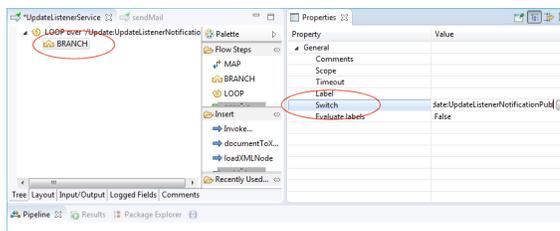
- 1 Open the flow service created in the previous step to add your custom logic.



- 2 Add a flow step **BRANCH** within the **LOOP**. This enables you to decide on a specific action for a particular field.

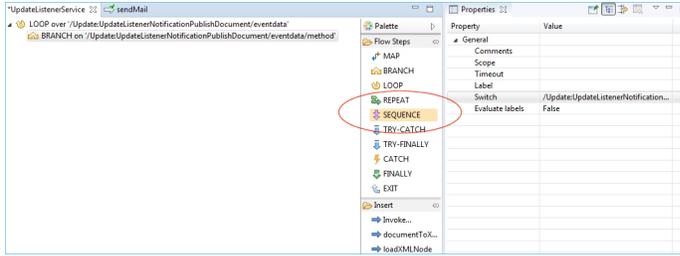


- 3 In the **Switch** property, set the path to an attribute of the event data. In this case it is the **method** attribute of the event data.

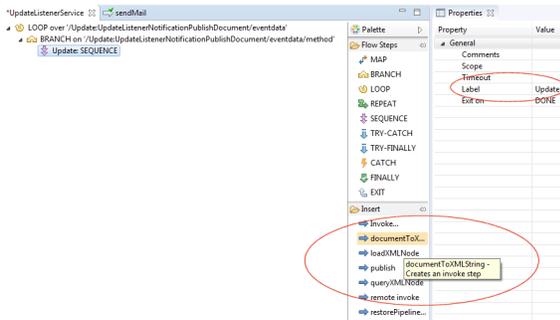


- 4 Under flow step **SEQUENCE**, specify what happens when the method has a particular value.

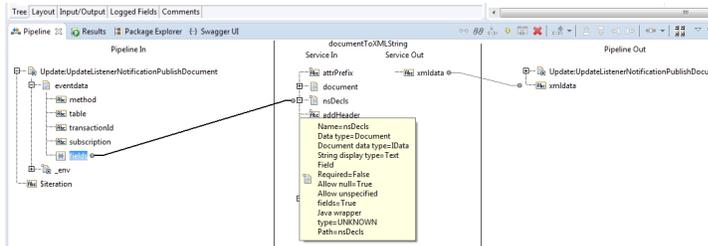
Scenario: Notifying a User after an Adabas Database Record is Updated



5 Set **Label** to **Update** and map document (the input format) to XML string.



6 Use the `pub.sml:documentToXMLString` service from the `WmPublic` package to create, from a document, an XML string that can be sent via email.



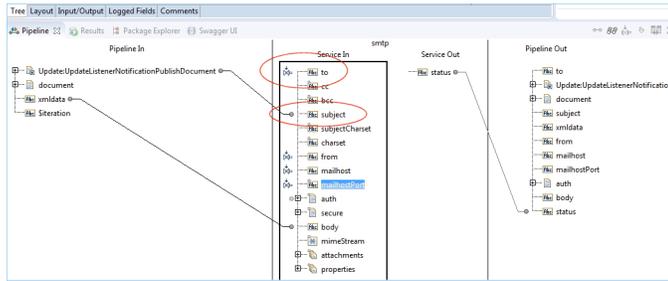
7 Specify the mail protocol, in our example "smtp". To send the mail you can use the public flow service `pub.client:smtp` from the `WmPublic` package.



8 Set the SMPT parameters.

- Some fields are coded manually, for example **to**. These fields are marked with the icon .
- Other fields are filled dynamically with values from the flow service, for example **subject**.

These fields are marked with the icon 



The result of these steps is an email where

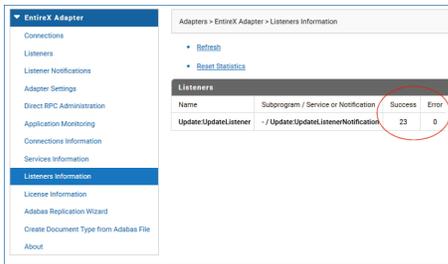
- the subject is dynamically filled with the value from the flow service
- the address is coded manually in a mapping operation
- the body is converted to XML and contains the fields that have been updated

```

EMPL
To: GER@eur.ad.sag
From: Do 01.09.2019 16:06
Cc: Gerth, Matthias

<?xml version="1.0"?>
<IDM>1128</IDM>
<CITY>Auckland</CITY>
<DEPT>SALED1</DEPT>
    
```


17 Troubleshooting



The screenshot shows the 'EntireX Adapter' management console. The left sidebar contains a menu with categories like 'Connections', 'Listeners', 'Adapter Settings', and 'Services Information'. The main area is titled 'Adapters > EntireX Adapter > Listeners Information'. It features a 'Refresh' button and a 'Reset Statistics' link. Below these is a table with the following data:

Name	Subprogram / Service or Notification	Success	Error
UpdateListener	/ UpdateListenerNotification	23	0

The 'Success' and 'Error' columns for the 'UpdateListener' row are circled in red in the original image.

