Informatica B2B Data Exchange
(Version 10.0)

Developer Guide

This product includes software licensed under the Academic Free License (http://www.opensource.org/licenses/afl-3.0.php), the Common Development and Distribution License (http://www.opensource.org/licenses/2df-1.0.php) the Sun Binary Code License Agreement Supplemental Terms, the BSD License (http://www.opensource.org/licenses/bsd-license.php), the new BSD License (http://opensource.org/licenses/bsd-3-Clause), the MIT License (http://www.opensource.org/licenses/mit-license.php), the Artistic License (http://www.opensource.org/licenses/artistic-license-1.0) and the Initial Developer's Public License Version 1.0 (http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/).

This product includes software copyright © 2003-2006 Joe Walnes, 2006-2007 XStream Committers. All rights reserved. Permissions and limitations regarding this software are subject to terms available at http://xstream.codehaus.org/license.html. This product includes software developed by the Indiana University Extreme! Lab. For further information please visit http://www.extreme.indiana.edu/.

This product includes software Copyright (c) 2013 Frank Balluff and Markus Moeller. All rights reserved. Permissions and limitations regarding this software are subject to the MIT license.


DISCLAIMER: Informatica LLC provides this documentation “as is” without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the “Software” includes certain drivers (the “DataDirect Drivers”) from DataDirect Technologies, an operating company of Progress Software Corporation (“DataDirect”) which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.

2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

Part Number: DX-DVG-96200-0001
# Table of Contents

**Preface** ................................................................. 9  
Informatica Resources .................................................. 9  
Informatica My Support Portal ......................................... 9  
Informatica Documentation ............................................. 9  
Informatica Product Availability Matrixes ......................... 9  
Informatica Web Site .................................................... 10  
Informatica How-To Library ........................................... 10  
Informatica Knowledge Base ........................................... 10  
Informatica Support YouTube Channel ............................. 10  
Informatica Marketplace ............................................... 10  
Informatica Velocity .................................................... 10  
Informatica Global Customer Support ............................... 10

**Chapter 1: PowerCenter Integration** ............................. 12  
PowerCenter Integration Overview ..................................... 12  
Developing Custom Mappings ........................................... 12  
Developing Custom Mappings Process ................................ 13  
Step 1. Create the Source and Target Definitions ................ 13  
Step 2. Create the Mapping ............................................. 16  
Step 3. Configure the JMS and JNDI Connections ................. 16  
Step 4. Create the PowerCenter Workflow and Session .......... 17  
Step 5. Export the PowerCenter Workflow and Mapping ....... 18

**Chapter 2: B2B Data Exchange Transformations** ............. 19  
B2B Data Exchange Transformations Overview .................... 19  
Installing and Registering Transformations ....................... 20  
Configuring Transformations ......................................... 21  
Handling Transformation Errors ...................................... 21  
B2B Data Exchange Transformations Rules and Guidelines ..... 21  
DX_Add_Document_To_Event Transformation ........................ 22  
Input Ports .................................................................... 22  
Input/Output Ports ....................................................... 23  
B2B Data Exchange Properties ....................................... 24  
DX_Aggregate Transformation ......................................... 24  
Input Ports .................................................................... 25  
Input/Output Ports ....................................................... 25  
B2B Data Exchange Properties ....................................... 26  
DX_Complete_Correlation Transformation ........................ 26  
Input Ports .................................................................... 26  
Input/Output Ports ....................................................... 27
Chapter 10: B2B Data Exchange Forms Designer Java API......................... 98
B2B Data Exchange Forms Designer Java API Overview. .......................... 98
B2B Data Exchange Forms Designer Java API Reference. ........................ 99

Index. .................................................................................................. 100
Preface

The B2B Data Exchange Developer Guide provides information about the tasks required to develop workflows in PowerCenter to process B2B Data Exchange documents. It assumes that you have a working knowledge of PowerCenter and are familiar with the format and requirements of the documents processed in B2B Data Exchange. The Dashboard and Reports topics assume you have a working knowledge of Logi Info Studio and the process of creating and deploying custom Dashboard panels and reports.

Informatica Resources

Informatica My Support Portal

As an Informatica customer, the first step in reaching out to Informatica is through the Informatica My Support Portal at https://mysupport.informatica.com. The My Support Portal is the largest online data integration collaboration platform with over 100,000 Informatica customers and partners worldwide.

As a member, you can:

- Access all of your Informatica resources in one place.
- Review your support cases.
- Search the Knowledge Base, find product documentation, access how-to documents, and watch support videos.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Documentation

The Informatica Documentation team makes every effort to create accurate, usable documentation. If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com. We will use your feedback to improve our documentation. Let us know if we can contact you regarding your comments.

The Documentation team updates documentation as needed. To get the latest documentation for your product, navigate to Product Documentation from https://mysupport.informatica.com.

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. You can access the PAMs on the Informatica My Support Portal at https://mysupport.informatica.com.
Informatica Web Site
You can access the Informatica corporate web site at https://www.informatica.com. The site contains information about Informatica, its background, upcoming events, and sales offices. You will also find product and partner information. The services area of the site includes important information about technical support, training and education, and implementation services.

Informatica How-To Library
As an Informatica customer, you can access the Informatica How-To Library at https://mysupport.informatica.com. The How-To Library is a collection of resources to help you learn more about Informatica products and features. It includes articles and interactive demonstrations that provide solutions to common problems, compare features and behaviors, and guide you through performing specific real-world tasks.

Informatica Knowledge Base
As an Informatica customer, you can access the Informatica Knowledge Base at https://mysupport.informatica.com. Use the Knowledge Base to search for documented solutions to known technical issues about Informatica products. You can also find answers to frequently asked questions, technical white papers, and technical tips. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team through email at KB.Feedback@informatica.com.

Informatica Support YouTube Channel
You can access the Informatica Support YouTube channel at http://www.youtube.com/user/INFASupport. The Informatica Support YouTube channel includes videos about solutions that guide you through performing specific tasks. If you have questions, comments, or ideas about the Informatica Support YouTube channel, contact the Support YouTube team through email at supportvideos@informatica.com or send a tweet to @INFASupport.

Informatica Marketplace
The Informatica Marketplace is a forum where developers and partners can share solutions that augment, extend, or enhance data integration implementations. By leveraging any of the hundreds of solutions available on the Marketplace, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at http://www.informaticamarketplace.com.

Informatica Velocity
You can access Informatica Velocity at https://mysupport.informatica.com. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Global Customer Support
You can contact a Customer Support Center by telephone or through the Online Support. Online Support requires a user name and password. You can request a user name and password at http://mysupport.informatica.com.
CHAPTER 1

PowerCenter Integration

This chapter includes the following topics:

- PowerCenter Integration Overview, 12
- Developing Custom Mappings, 12

PowerCenter Integration Overview

PowerCenter is a tool you use to process B2B Data Exchange documents. You use the PowerCenter Client tools to develop workflows that process B2B Data Exchange documents.

You can find sample workflows in the following directory: <DXInstallationDir>/samples. Each sample workflow has an associated readme file that describes the sample workflow and contains instructions.

Developing Custom Mappings


Before you develop B2B Data Exchange workflows in PowerCenter, verify that the B2B Data Exchange PowerCenter client and server plug-ins are installed and registered to the PowerCenter repository. For details, see the B2B Data Exchange Installation and Configuration Guide.

After you develop and export the workflow and the mapping, you use the Operation Console to import the XML file as the workflow definition file into the B2B Data Exchange repository.

When you add PowerCenter transformations to a PowerCenter mapping, you can add workflow parameters that the corresponding B2B Data Exchange workflow can use. You can use the Forms Designer, in the B2B Data Exchange Operation Console, to customize the layout and behavior of the workflow parameters.

If you edit the PowerCenter workflow or mapping, you must update the B2B Data Exchange workflow by re-selecting the PowerCenter workflow or the XML file in the B2B Data Exchange Operation Console. During the update process, you can resolve changes to workflow parameters and parameter types. When you change the type of a workflow parameter type, the import process deletes the values for the imported parameters and you must manually enter the parameter values.
Developing Custom Mappings Process

To develop a workflow in PowerCenter that processes B2B Data Exchange data, perform the following steps:

1. Create the source and target definitions.
2. Create the mapping and add transformations.
3. Configure the PowerExchange JNDI and JMS connections.
4. Create the workflow and the session.
5. Export the PowerCenter workflow and mapping to an XML file.

Step 1. Create the Source and Target Definitions

B2B Data Exchange uses a JMS broker to send and receive documents to and from PowerCenter. The documents can be text or binary documents.

When you use a real-time workflow, you add a JMS source definition to a mapping that receives and processes B2B Data Exchange documents. If you send documents to B2B Data Exchange, you must also define a JMS target. PowerCenter batch workflows can have a flat file source or no source at all, and run with the PowerCenter Web Services Hub.

You create the source and target definitions in the PowerCenter Designer in the same way that you create source and target definitions for other mappings. For general information about source and target definitions, see the PowerCenter Designer Guide.

The source and target definitions require additional properties to allow B2B Data Exchange to pass parameters to and from PowerCenter. Batch workflows require additional mapping parameters that allow B2B Data Exchange to send data to and from with PowerCenter.

Creating the Source Definition

In the PowerCenter Designer, create the source and define the source properties.

The source definition process includes the following steps:

1. Create the source object.
2. Add ports to the source object. Add port information as properties in the JMS Message Property Columns tab of the source definition. By default, the port properties have the prefix Prop_.

You can filter data to consume in the Source Filter attribute of the Source Qualifier transformation in subscription workflow. The following example shows the field syntax to filter by ID and date range in a single line:

```
MY_TABLE.DIH__PUBLICATION_INSTANCE_ID in ($$myTopic__PublicationInstanceIDs)
AND MY_TABLE.DIH__PUBLICATION_INSTANCE_DATE in ($$myTopic__PublicationInstanceDatesSQL)
```
The following table describes the B2B Data Exchange source properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accountName</td>
<td>Name of the account for which the B2B Data Exchange server sets the port that the workflow uses to process the event. Note: The property value can be null if you use a global profile.</td>
</tr>
<tr>
<td>accountNumber</td>
<td>Number of the account for which the B2B Data Exchange server sets the port that the workflow uses to process the event. The property value must be a string. Note: The property value can be null if you use a global profile.</td>
</tr>
<tr>
<td>application</td>
<td>Name of the application for which the B2B Data Exchange server sets the port that the workflow uses to process the event.</td>
</tr>
<tr>
<td>eventId</td>
<td>Identifier of the associated event for the documents that the workflow processes.</td>
</tr>
<tr>
<td>isEventReprocessed</td>
<td>Determines if the associated event is reprocessed.</td>
</tr>
<tr>
<td>partnerName</td>
<td>Name of the partner for which the B2B Data Exchange server sets the port that the workflow uses to process the event. Note: The property value can be null if you use a global profile.</td>
</tr>
<tr>
<td>partnerId</td>
<td>Identifier of the partner for which the B2B Data Exchange server sets the port that the workflow uses to process the event. Note: The property value can be null if you use a global profile.</td>
</tr>
<tr>
<td>profileId</td>
<td>Identifier of the profile for which the B2B Data Exchange server sets the port that the workflow uses to process the event.</td>
</tr>
<tr>
<td>profileName</td>
<td>Name of the profile for which the B2B Data Exchange server sets the port that the workflow uses to process the event.</td>
</tr>
<tr>
<td>DXSentEventId</td>
<td>List of identifiers for the events that triggered the event monitor. Available when a regular delivery or a batch delivery event monitor triggers the workflow.</td>
</tr>
<tr>
<td>DXMonitorId</td>
<td>Identifier for the event monitor that triggered the workflow. Available when a regular delivery or a batch delivery event monitor triggers the workflow.</td>
</tr>
<tr>
<td>DXMonitorName</td>
<td>Name of the event monitor that triggered the workflow. Available when a regular delivery or a batch delivery event monitor triggers the workflow.</td>
</tr>
<tr>
<td>DXUser</td>
<td>Name of the user that received the monitor notification. Available when a regular delivery event monitor triggers the workflow.</td>
</tr>
</tbody>
</table>
Creating the Target Definition

In the PowerCenter Designer, create the target and add ports to store properties for running the workflow. You can also add properties to store other B2B Data Exchange values that you want to send back from PowerCenter.

The target definition process includes the following steps:

1. Create the target object.
2. Add ports to the target object. Add the port information as properties in the JMS Message Property Columns tab of the target definition. By default, the port properties have the prefix `Prop_`.

The following table describes the B2B Data Exchange target properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| DXDataByReference | Required. Determines whether the JMS message that PowerCenter sends back to B2B Data Exchange contains a document reference. Possible values:  
- True. The JMS message contains a reference to a document that is stored in the B2B Data Exchange document store.  
- False. The JMS message contains the actual document data. |
| DXEventId         | Identifier of the event to use in the next workflow. If you do not set the value of this property, the B2B Data Exchange server generates a new event ID for the next workflow. |
| DXParentEventId   | Identifier of the parent event to use in the next workflow. Set the value of this property if you want to send the JMS message to multiple targets with a child event for each target. The B2B Data Exchange then generates a new event ID for the next workflow to use for the child events. Note: If you do not set the value of this property and send the message to multiple targets, all targets inherit the parent event status and you cannot track the status of each child event. |
| DXStatusName      | Name of the event status to assign to the event in B2B Data Exchange after a successful processing. |
| endpointName      | Name of the endpoint to which to route the message after processing. |
| profileId         | Identifier of the profile to which to route the message after processing. |
| profileName       | Name of the profile to which to route the message after processing. |
| accountName       | Name of the account to which to route the message after processing. |
| partnerName       | Name of the partner to which to route the message after processing. |
| sendToAccount     | Determines whether to send the message to a global endpoint or to an endpoint that is associated with an account or a partner. Possible values:  
- True. Sends the message to an endpoint that is associated with an account or a partner. If you set this value, you must set a value for the `accountName` or `partnerName` properties.  
- False. Sends the message to a global endpoint. |
Step 2. Create the Mapping

Create a mapping that contains the source definition, target definition, and transformations that you want to use to process documents for B2B Data Exchange.

You create a mapping for B2B Data Exchange in the same way you build other PowerCenter mappings. Use B2B Data Exchange transformations and Unstructured Data transformations to add product-specific functionality to the mapping. You can also add other PowerCenter transformation to the mapping.

If you require workflow parameters, add the DX_Profile_Parameters transformation. The transformation gets profile parameter values during run-time.

Step 3. Configure the JMS and JNDI Connections

Create a PowerExchange JNDI application connection for the B2B Data Exchange JMS provider and a JMS application connection for the JMS source.

Before you create the connection objects, verify that PowerExchange for JMS is installed on the same machine as PowerCenter and that you can create JMS connections. For general information about PowerExchange for JMS, see the PowerExchange for JMS User Guide for PowerCenter.

JMS sources and targets require the following PowerExchange connection objects:

- JMS application connection. Specific connection information for the JMS source.

In each connection object, you configure specific attributes to enable PowerCenter to send and receive messages from B2B Data Exchange.

JNDI Connection Attributes

You define the JNDI application connection in PowerCenter and configure specific attributes to connect to the B2B Data Exchange JMS provider.

The following table describes the attributes of the JNDI connection object to configure:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JNDI Context Factory</td>
<td>Name of the context factory for the B2B Data Exchange JMS provider. Set the attribute to the following value: com.informatica.b2b.dx.jndi.DXContextFactory</td>
</tr>
<tr>
<td>JNDI Provider URL</td>
<td>URL for the JNDI provider in B2B Data Exchange. The host name and the port number must match the host name and the port number in the jndiProviderURL attribute of the JMS endpoints in the B2B Data Exchange configuration file. If you use a single node installation, the default JNDI provider URL is: failover:tcp://localhost:18616 If you use a cluster configuration, you can add multiple hosts.</td>
</tr>
</tbody>
</table>
**JMS Connection Attributes**

You define the JMS application connection in PowerCenter and configure specific attributes to connect to the input queue of the JMS source. You create a connection for each workflow. The input queue configuration must match the B2B Data Exchange workflow that represents the PowerCenter workflow.

All output queues of the JMS targets in B2B Data Exchange workflows can use the same JMS application connection. The output queue configuration can use the default inbound JMS endpoint that you define in the B2B Data Exchange configuration file.

The following table describes the attributes of the JMS connection object to configure:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMS Destination Type</td>
<td>Type of JMS destination for the Data Exchange messages. Set the attribute to the following value: QUEUE</td>
</tr>
<tr>
<td>JMS Connection Factory Name</td>
<td>Name of the connection factory in the JMS provider. Set the attribute to the following value: connectionfactory.local</td>
</tr>
<tr>
<td>JMS Destination</td>
<td>Name of the destination. The destination value must match the name of the B2B Data Exchange workflow that represents the PowerCenter workflow. Set the attribute value in the following format: queue.&lt;DXWorkflowName&gt;</td>
</tr>
</tbody>
</table>

**Step 4. Create the PowerCenter Workflow and Session**

Create the workflow and the session that runs the mapping. You create the workflow in the same way you create other PowerCenter workflows. If you use a JMS source, you must configure the session as a real-time session.

1. In the Workflow Designer, create the workflow and the session object. For general information about creating workflows and sessions, see the PowerCenter Advanced Workflow Guide.
2. In the Edit Tasks dialog box, click the Properties tab.
3. Set the values for the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commit Type</td>
<td>Source</td>
</tr>
<tr>
<td>Commit Interval</td>
<td>1</td>
</tr>
</tbody>
</table>

4. In the Properties section for the selected source type, define the source attributes.

Set values for the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source filename</td>
<td>$InputFile_&lt;eventAttributeValue&gt;</td>
</tr>
<tr>
<td>Note:</td>
<td>The event attribute value must match the value in the source event that B2B Data Exchange sends to the workflow.</td>
</tr>
</tbody>
</table>
• If you select a JMS source, set the values for the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time Flush Latency</td>
<td>1</td>
</tr>
<tr>
<td>JMS Queue Reader Mode</td>
<td>Message Consumer</td>
</tr>
</tbody>
</table>

• If you select a file source, set the values for the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source filename</td>
<td>$InputFile_&lt;eventAttributeValue&gt;</td>
</tr>
</tbody>
</table>

*Note:* The event attribute value must match the value in the source event that B2B Data Exchange sends to the workflow.

5. Add the transformations that notify the B2B Data Exchange server that the publication or that the subscription process ended.
   • If you develop a publication workflow, add the DX_Notification transformation to trigger the subscription workflows and change the publication event status.
   • If you develop a subscription workflow, add the DX_Event_Details to change the subscription event status.

6. Save the session object.

7. To choose the format of the JMS message, select the **Message Body Type** option in the **JMS Message Body Columns** tab of the **Edit Tables** dialog box.
   • TextMessage. Sends the JMS message as a plain text file.
   • BytesMessage. Sends the JMS message as a binary file.
   *Note:* You cannot use the MapMessage format in workflows for B2B Data Exchange.

8. Test the workflow to ensure that it works correctly.

### Step 5. Export the PowerCenter Workflow and Mapping

Export the workflow and the mapping from PowerCenter to an XML file. You then import the XML file to B2B Data Exchange. You must use the Repository Manager to export the mapping and the workflow.

You create a B2B Data Exchange workflow in the Operation Console. The B2B Data Exchange workflow represents the PowerCenter workflow and mapping. When you create the B2B Data Exchange workflow, select the XML file as the workflow definition file to import the PowerCenter workflow to the B2B Data Exchange repository.

You can use the Forms Designer to customize the layout and behavior of the workflow parameters that appear when the operator creates or edits the profile that runs the workflow.

If you edit the PowerCenter workflow or mapping, you must export and import the XML file to B2B Data Exchange. During the import process, you can resolve changes to workflow parameters and parameter types. When you change the type of a workflow parameter type, the import process deletes the values for the imported parameters and you must manually enter the parameter values.

When you import an updated workflow definition file, a list of dependent profiles appear. You can disable dependent profiles until the operator reviews the profiles.
B2B Data Exchange Transformations

This chapter includes the following topics:

- [B2B Data Exchange Transformations Overview, 19](#)
- [DX_Add_Document_To_Event Transformation, 22](#)
- [DX_Aggregate Transformation, 24](#)
- [DX_Complete_Correlation Transformation, 26](#)
- [DX_Create_Event Transformation, 27](#)
- [DX_Event_Attribute Transformation, 29](#)
- [DX_Event_Details Transformation, 30](#)
- [DX_Generate_Temporary_File Transformation, 31](#)
- [DX_Get_Document_File_Path Transformation, 32](#)
- [DX_Increment_Profile_Parameter Transformation, 33](#)
- [DX_Initiate_Correlation Transformation, 34](#)
- [DX_Profile_Parameters Transformation, 35](#)
- [DX_Release_Delayed_Events Transformation, 37](#)
- [DX_ResOLVE_Profile, 37](#)
- [DX_Throw_Error, 39](#)

B2B Data Exchange Transformations Overview

A B2B Data Exchange transformation is a set of functions that process B2B Data Exchange documents in PowerCenter.

When you install B2B Data Exchange, you can install a set of transformations that you use in PowerCenter workflows to process B2B Data Exchange documents. When you create mappings, you can use the transformations to perform the functions that you require in the same way you use other PowerCenter transformations.

B2B Data Exchange transformations are custom Java transformations that you use to access the B2B Data Exchange API without writing Java code. You can use these transformations, the Unstructured Data transformation, and other transformations to process B2B Data Exchange documents.
You can also create custom transformations and use the methods in the B2B Data Exchange Web Services API to define the transformation behavior.

The following table describes the B2B Data Exchange transformations:

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DX_Add_Document_To_Event</td>
<td>Attaches a document to an event.</td>
</tr>
<tr>
<td>DX_Aggregate</td>
<td>Groups rows of data.</td>
</tr>
<tr>
<td>DX_Complete_Correlation</td>
<td>Completes the correlation for an event.</td>
</tr>
<tr>
<td>DX_Create_Event</td>
<td>Creates an event and sets the properties of the event.</td>
</tr>
<tr>
<td>DX_Event_Attribute</td>
<td>Gets or sets the values of event attributes.</td>
</tr>
<tr>
<td>DX_Event_Details</td>
<td>Gets or sets the values of a property of an event.</td>
</tr>
<tr>
<td>DX_Generate_Temporary_File</td>
<td>Generates a file name for a file in the B2B Data Exchange document store.</td>
</tr>
<tr>
<td>DX_Get_Document_File_Path</td>
<td>Converts a document reference into a file path.</td>
</tr>
<tr>
<td>DX_Increment_Profile_Parameter</td>
<td>Increments the value of a parameter associated with a profile.</td>
</tr>
<tr>
<td>DX_Initiate_Correlation</td>
<td>Initiates a correlation for an event.</td>
</tr>
<tr>
<td>DX_Profile_Parameters</td>
<td>Gets profile parameter values.</td>
</tr>
<tr>
<td>DX_Release_Delayed_Events</td>
<td>Releases all events for a profile.</td>
</tr>
<tr>
<td>DX_Resolve_Profile</td>
<td>Determines the profile ID based on the application, account number, and partner.</td>
</tr>
<tr>
<td>DX_Throw_Error</td>
<td>Sets an event status to error if the transformation fails.</td>
</tr>
</tbody>
</table>

**Installing and Registering Transformations**

To add B2B Data Exchange transformations to PowerCenter, use the B2B Data Exchange installer.

You install the following components:

- B2B Data Exchange PowerCenter server plug-in. Use the Administrator tool to register the plug-in.
- B2B Data Exchange PowerCenter Client plug-in. The installer registers the plug-in during the installation process.

After you install and register the plug-ins, verify that the transformations are enabled in the PowerCenter Designer. For more information about installing and configuring the plug-ins, see the *B2B Data Exchange Installation and Configuration Guide*. 
Configuring Transformations

After you add a B2B Data Exchange transformation to a mapping, you edit and configure the transformation based on your document processing requirements.

When you edit a B2B Data Exchange transformation, you can configure the transformation components on the following tabs:

• Transformation tab. Rename the transformation and add a description.
• Ports. Add, edit, or delete ports. You can also designate ports as input or output ports.
• Properties tab. Configure transformation properties, such as module and function identifiers, transaction properties, and the runtime location. The properties that appear on this tab are the same as the properties for other PowerCenter Custom transformations. For more information about the Properties tab, see the PowerCenter Transformation Guide.
• DX Properties tab. Set the default values for the ports in the transformation. You can also set the default values for other B2B Data Exchange transformation properties.

Handling Transformation Errors

You can use certain transformation ports to define how to handle errors that occur during a workflow run.

Each B2B Data Exchange transformation uses the following ports to handle errors:

• DXErrorCode. When a transformation fails, the transformation sets the DXErrorCode to a value greater than zero.
• DXErrorMessage. When a transformation fails, the transformation sends an error message that describes the failure to the DXErrorMessage port.

When a transformation generates an error, the transformation writes the error to the PowerCenter session log. The error log includes the exception class, description, cause, and stack trace. The logging level is based on the PowerCenter configuration. Up to 1K of the document associated with the error will be included in the log.

If the option to set the event status to error when a transformation fails is set to true, the transformation also sets the status of the event to error.

You can set the error handling ports as input ports to prevent the transformation from running if an input error occurs.

B2B Data Exchange Transformations Rules and Guidelines

When you work with B2B Data Exchange transformations in PowerCenter, follow rules and guidelines to optimize performance and prevent errors.

The following list describes rules and guidelines for working with B2B Data Exchange transformations:

• B2B Data Exchange transformations are based on the PowerCenter Custom transformation and they provide the same configuration options as other custom transformations. You can use them as you use other PowerCenter transformations.
• B2B Data Exchange transformations are connected transformations. Connected transformations pass data to or from other transformations.
• B2B Data Exchange transformations are passive transformations except for the DX_Aggregate transformation.
• If a transformation port has a corresponding B2B Data Exchange property, the value of the port takes precedence over the value of the property during runtime. When the session runs, if the value of the port
is not null, the PowerCenter Integration Service uses the value of the port for processing. If the value of the port is null, the PowerCenter Integration Service uses the value of the B2B Data Exchange property for processing.

- Port names are case insensitive and prefix insensitive. DXEventID, dxEVENTid, and eventid are all treated as the same port.
- You can process only one event ID with one transformation at a time. To simultaneously process an event with different transformations, use the DX_Create_Event transformation to create a child event for each additional transformation that you want to use.
- When you run a PowerCenter workflow that uses a B2B Data Exchange transformation, PowerCenter tries to connect to the B2B Data Exchange repository to get the list of event statuses and types to use in the transformation. An error message indicates a failed connection. If the connection fails, PowerCenter gets the event type values from the Client plug-in configuration file. To resolve connection errors, verify that the connection section in the dxplugin.ini file contains the following configuration:

```
[DX_REPOSITORY]
; ODBC connection string to the DX repository
; CONNECTION_STRING=DRIVER={DataDirect 7.0 Oracle Wire Protocol};UID=%1;PWD=%2;Host=localhost;Port=1521;SN=orcl
; CUSTOM_CONNECTION_STRING
; ODBC DSN to the DX repository
; DSN NAME=dxOdbcResourceName
USER_NAME=DX
USER_PASSWD=DX
EVENT_TYPE_NAME=SELECT event_type_name FROM dx_event_type ORDER BY 1
EVENT_STATUS_NAME=SELECT event_status_name FROM dx_event_status ORDER BY 1
```

**DX_Add_Document_To_Event Transformation**

The DX_Add_Document_To_Event transformation attaches a document to an event.

You can use the transformation to add a document that you create in previous transformations. For example, you can use the transformation to attach a log file to an event.

**Input Ports**

Configure input ports for the DX_Add_Document_To_Event transformation on the Ports tab.

The following table describes the DX_Add_Document_To_Event input ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXDescription</td>
<td>string</td>
<td>Description of the document to attach to the event.</td>
</tr>
<tr>
<td>DXMIMEType</td>
<td>string</td>
<td>MIME type of the document to attach to the event.</td>
</tr>
<tr>
<td>Port</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DXEncoding</td>
<td>string</td>
<td>Character encoding of the document to attach to the event. This is the character set to use to convert strings to byte arrays.</td>
</tr>
<tr>
<td>DXTemporaryFilePath</td>
<td>string</td>
<td>Optional. Path and file name generated by the DX_Generate_Temporary_File transformation where the workflow stores the new file. The DX_Add_Document_To_Event transformation saves the file as a new document reference in the document store and attaches the file reference to the event. You can set this port or set the DXData and DXDataByReference ports. If this port and the DXData and DXDataByReference ports are not set, the transformation creates an empty document and adds it to the event.</td>
</tr>
</tbody>
</table>

### Input/Output Ports

Configure input/output ports for the DX_Add_Document_To_Event transformation on the Ports tab.

The following table describes the DX_Add_Document_To_Event input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXEventId</td>
<td>string</td>
<td>Required. Identifier of the event to which to attach the document.</td>
</tr>
<tr>
<td>DXData</td>
<td>string</td>
<td>Document to attach to the event. This port can contain the data of the document or a reference to the document. If this port is null, the transformation creates an empty document and adds the document to the event. To attach a document with text data, set the datatype of the port to string or text. To attach a document with binary data, set the datatype of the port to binary.</td>
</tr>
<tr>
<td>DXDataByReference</td>
<td>string</td>
<td>Indicates whether the DXData port contains the document data or a document reference. If the value is true, the DXData port contains a document reference. If the value is null or false, the DXData port contains the document data.</td>
</tr>
<tr>
<td>DXDocumentId</td>
<td>string</td>
<td>Identifier of the document to attach to the event.</td>
</tr>
<tr>
<td>DEXErrorMessage</td>
<td>string</td>
<td>Error message generated by the transformation.</td>
</tr>
<tr>
<td>DEXErrorCode</td>
<td>string</td>
<td>Error code generated by the transformation. If the transformation fails, the value of the DEXErrorCode port is greater than zero.</td>
</tr>
</tbody>
</table>
B2B Data Exchange Properties

Configure properties for the DX_Add_Document_To_Event transformation on the DX Properties tab.

The following table describes the B2B Data Exchange properties of the DX_Add_Document_To_Event transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Description of the document to attach to the event.</td>
</tr>
<tr>
<td>Document Role</td>
<td>Role of the document when it is attached to an event.</td>
</tr>
<tr>
<td></td>
<td>Select one of the following roles:</td>
</tr>
<tr>
<td></td>
<td>- SOURCE. The document is attached to an event as a source document.</td>
</tr>
<tr>
<td></td>
<td>When you view the event in the Operation Console, you can view the</td>
</tr>
<tr>
<td></td>
<td>attached document in the Input section.</td>
</tr>
<tr>
<td></td>
<td>- TARGET. The document is attached to an event as a target document.</td>
</tr>
<tr>
<td></td>
<td>When you view the event in the Operation Console, you can view the</td>
</tr>
<tr>
<td></td>
<td>attached document in the Output section.</td>
</tr>
<tr>
<td></td>
<td>- LOG. The document is attached to an event as a log document.</td>
</tr>
<tr>
<td></td>
<td>When you view the event in the Operation Console, you can view the</td>
</tr>
<tr>
<td></td>
<td>attached document in the Logging Information section.</td>
</tr>
<tr>
<td>Generate an error in this transformation</td>
<td>Indicates whether to set the status of the event to error when the</td>
</tr>
<tr>
<td></td>
<td>transformation generates an error.</td>
</tr>
<tr>
<td></td>
<td>Selected by default.</td>
</tr>
</tbody>
</table>

DX_Aggregate Transformation

The DX_Aggregate active transformation groups segments of data to process.

You can use the transformation to aggregate row sets that you previously split back into a single document.

You set the DXParentEventId and the DXGroupByKey ports to sort the data that you want to aggregate.

The DX_Aggregate transformation groups rows in the following manner:

1. When the transformation reads the first row in the group, it creates a temporary file and writes the content of the DXData port to the file.
2. On each subsequent row, the transformation appends the DXDelimiter and the content of the DXData port to the file. It then increments the number of rows in the DXRowCount port.
3. When the transformation reads the last row in the group, it closes the temporary file and increments the number of groups in the DXGroupSequence port. It then passes the output to the pipeline.
4. If you use a pass-through port, the transformation sends the value from the last row of the group to the pipeline.

If a row contains an error or a null value in the DXData port, the transformation drops the row from the file and does not pass the output to the pipeline.

The DX_Aggregate transformation determines the end of the group in the following cases:

- There is no more data to aggregate.
- The value of the DXParentEventId port differs from the value in the previous row.
- The value of the DXGroupByKey port differs from the value in the previous row.
• The row count for the group reached the value that you define in the DXMaxRowsInGroup port.
• The value of the DXLastInGroup port is true.

Input Ports

Configure input ports for the DX_Aggregate transformation on the Ports tab.

The following table describes the DX_Aggregate input ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXParentEventId</td>
<td>string</td>
<td>Identifier of the parent event associated with the data. This is the primary GROUP BY key.</td>
</tr>
<tr>
<td>DXGroupByKey</td>
<td>string</td>
<td>Indicates whether to group rows with the primary key.</td>
</tr>
<tr>
<td>DXDelimiter</td>
<td>string</td>
<td>Delimiter string to add between segments of data. The value determines the end of a segment and the start of another segment.</td>
</tr>
<tr>
<td>DXData</td>
<td>string binary</td>
<td>Data segment to aggregate. This port can contain the data segment or a document reference.</td>
</tr>
<tr>
<td>DXDataByReference</td>
<td>string</td>
<td>Indicates whether the DXData port contains data or a document reference. If the value is true, the DXData port contains a document reference. If the value is null or false, the DXData port contains data.</td>
</tr>
<tr>
<td>DXLastInGroup</td>
<td>string</td>
<td>Indicates whether the row is the last row in a group.</td>
</tr>
<tr>
<td>DXMaxRowsInGroup</td>
<td>integer</td>
<td>Maximum number of rows in a group.</td>
</tr>
</tbody>
</table>

Input/Output Ports

Configure input/output ports for the DX_Aggregate transformation on the Ports tab.

The following table describes the DX_Aggregate input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXGroupSequence</td>
<td>integer</td>
<td>Number of groups the transformation processed since the PowerCenter Integration Service started. The transformation increments the value of this port every time a group is processed. When the PowerCenter Integration Service starts, the value is set to zero.</td>
</tr>
<tr>
<td>DXRowCount</td>
<td>integer</td>
<td>Number of rows successfully processed and added to the group.</td>
</tr>
<tr>
<td>DXTemporaryFilePath</td>
<td>string</td>
<td>Path and file name for the temporary file where the transformation stores the grouped data. You can use this temporary file in subsequent transformations to create a document and add it to an event.</td>
</tr>
<tr>
<td>DXErrorCount</td>
<td>integer</td>
<td>Number of rows that generated errors and were not added to the group.</td>
</tr>
</tbody>
</table>
### DXErrorMessage

**Type:** string  
**Description:** Error message generated by the transformation.

### DXErrorCode

**Type:** string  
**Description:** Error code generated by the transformation. If the transformation fails, the value of the DXErrorCode port is greater than zero.

## B2B Data Exchange Properties

Configure properties for the DX_Aggregate transformation on the **DX Properties** tab.

The following table describes the B2B Data Exchange properties of the DX_Aggregate transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| Generate an error in case a failure occurs in this transformation | Indicates whether to set the status of the event to error when the transformation generates an error.  
Selected by default. |

## DX_Complete_Correlation Transformation

The DX_Complete_Correlation transformation completes a correlation for an event.

Use the transformation with the DX_Initiate_Correlation. The DX_Initiate_Correlation generates a correlation ID that you must pass to the DX_Complete_Correlation to complete the correlation.

### Input Ports

Configure input ports for the DX_Complete_Correlation transformation on the **Ports** tab.

The following table describes the DX_Complete_Correlation input ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXCorrelationId</td>
<td>string</td>
<td>Required. Identifier of the correlation to complete.</td>
</tr>
</tbody>
</table>
Input/Output Ports

Configure input/output ports for the DX_Complete_Correlation transformation on the Ports tab. The following table describes the DX_Complete_Correlation input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXEventId</td>
<td>string</td>
<td>Required. Identifier of the event that completes the correlation.</td>
</tr>
<tr>
<td>DXErrorMessage</td>
<td>string</td>
<td>Error message generated by the transformation.</td>
</tr>
<tr>
<td>DXErrorCode</td>
<td>string</td>
<td>Error code generated by the transformation. If the transformation fails, the value of the DXErrorCode port is greater than zero.</td>
</tr>
</tbody>
</table>

B2B Data Exchange Properties

Configure properties for the DX_Complete_Correlation transformation on the DX Properties tab. The following table describes the B2B Data Exchange properties of the DX_Complete_Correlation transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate an error if the reconciliation ID does not exist</td>
<td>Indicates whether to set the status of the event to error if the correlation ID does not exist. Selected by default.</td>
</tr>
<tr>
<td>Generate an error in case a failure occurs in this transformation</td>
<td>Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.</td>
</tr>
</tbody>
</table>

DX_Create_Event Transformation

The DX_Create_Event transformation creates an event in the Data Exchange repository and set the properties of the event.

If you define a parent event in the transformation, the workflow generates a child event for the parent event that you set.
Input Ports

Configure input ports for the DX_Create_Event transformation on the Ports tab.

The following table describes the DX_Create_Event input ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXParentEventId</td>
<td>string</td>
<td>Optional. Identifier of the parent event. If the DXParentEventId port has a value, the transformation creates a child event of the DXParentEventId event. If the DXParentEventId port is null and you select the Create root event property, the transformation creates a root event.</td>
</tr>
<tr>
<td>DXSubject</td>
<td>string</td>
<td>Subject of the event to create. The subject can contain up to 255 bytes.</td>
</tr>
<tr>
<td>DXStatusName</td>
<td>string</td>
<td>Status of the event to create. If the value is not null, the transformation sets the status for the event.</td>
</tr>
<tr>
<td>DXTypeName</td>
<td>string</td>
<td>Type of the event to create.</td>
</tr>
</tbody>
</table>

Input/Output Ports

Configure input/output ports for the DX_Create_Event transformation on the Ports tab.

The following table describes the DX_Create_Event input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXNewEventId</td>
<td>string</td>
<td>Identifier of the event to generate.</td>
</tr>
<tr>
<td>DXErrorMessage</td>
<td>string</td>
<td>Error message generated by the transformation.</td>
</tr>
<tr>
<td>DXErrorCode</td>
<td>string</td>
<td>Error code generated by the transformation. If the transformation fails, the value of the DXErrorCode port is greater than zero.</td>
</tr>
<tr>
<td>DXProfileId</td>
<td>string</td>
<td>Required. Identifier of the profile associated with the event to generate. If the profile ID port has a value, the transformation associates the profile ID with the event. If the profile ID port value is null, the transformation uses the DXApplication, DXAccountNumber, and DXPartnerName ports to determine the profile ID to associate with the event.</td>
</tr>
</tbody>
</table>

B2B Data Exchange Properties

Configure properties for the DX_Create_Event transformation on the DX Properties tab.

The following table describes the B2B Data Exchange properties of the DX_Create_Event transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Status</td>
<td>Status of the event to create.</td>
</tr>
<tr>
<td>Event Type</td>
<td>Type of the event to create.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Event Subject</td>
<td>Subject of the event to create.</td>
</tr>
<tr>
<td>Create root event</td>
<td>Indicates whether to create a root event. If true, the transformation creates a root event. If false, the transformation creates a child event of the DXParentEventId.</td>
</tr>
<tr>
<td>Generate an error in case a failure occurs in this transformation</td>
<td>Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.</td>
</tr>
</tbody>
</table>

**DX_Event_Attribute Transformation**

The DX_Event_Attribute gets or sets event attribute values.

To use this transformation, you first configure an event attribute in the Operation Console. You then add ports to the transformation. Each port represents the event attribute that you want to get or set.

The port name must match the event attribute name. However, the attribute name is not case sensitive. For example, the `event_att1` attribute and the `EVENT_ATT1` are treated as the same attribute.

Unlike other profile parameters, the event name is not prefix insensitive. For example, the `DX_event_att1` attribute and the `event_att1` attribute are not treated as the same attribute.

If you set a value for the port, the transformation sets the event attribute to this value. To get the value of an event attribute, create an empty port.

**Note:** The DXValue port, the DXAttributeKey property, and the Event attribute property are optional and retained for backwards compatibility. Use the `<attribute>` port.

**Input/Output Ports**

Configure input/output ports for the DX_Event_Attribute transformation on the **Ports** tab.

The following table describes the DX_Event_Attribute input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXEventId</td>
<td>string</td>
<td>Required. Identifier of the event associated with the attribute to get or to update.</td>
</tr>
<tr>
<td>DXErrorMessage</td>
<td>string</td>
<td>Error message generated by the transformation.</td>
</tr>
<tr>
<td>DXErrorCode</td>
<td>string</td>
<td>Error code generated by the transformation. If the transformation fails, the value of the DXErrorCode port is greater than zero.</td>
</tr>
</tbody>
</table>
B2B Data Exchange Properties

Configure properties for the DX_Event_Attribute transformation on the **DX Properties** tab.

The following table describes the B2B Data Exchange properties of the DX_Event_Attribute transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| Event attribute name                          | Name of the event attribute to update. You configure the property in the Operation Console. If the value is null, the transformation generates an error. 
  **Note:** Optional property. It is recommended that you use the `<Attribute>` port instead. This property is retained for backward compatibility. |
| Generate an error in case a failure occurs in this transformation | Indicates whether to set the status of the event to error when the transformation generates an error. 
  Selected by default. |

DX_Event_Details Transformation

The DX_Event_Details transformation gets or sets multiple properties for an event.

The transformation ports represent the properties of an event. The transformation sets the property of the event based on the value that the input port passes to the transformation. The transformation passes the value of the property through an output port to the event.

Input/Output Ports

Configure input/output ports for the DX_Event_Details transformation on the **Ports** tab.

The following table describes the DX_Event_Details input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| DXStatusName | string    | Status of the event. The status indicates the stages that the event passes during processing. 
  To set a status for a completed event that is different from Complete, set the status name in the DXStatusName attribute of the JMS Target definition property. |
| DXEventId    | string    | Required. Identifier of the event associated with the properties to get or update.                                                          |
| DXTypeName   | string    | Type of the event.                                                                                                                         |
| DXSubject    | string    | Subject of the event.                                                                                                                      |
| DXChildCount | double    | Number of child events associated with the event.                                                                                           |
| DXParentEventId | string   | Identifier of the parent event for this event.                                                                                             |
| DXErrorMessage | string  | Error message generated by the transformation.                                                                                             |
| DXErrorCode  | string    | Error code generated by the transformation. If the transformation fails, the value of the DXErrorCode port is greater than zero.         |
B2B Data Exchange Properties

Configure properties for the DX_Event_Details transformation on the DX Properties tab.

The following table describes the B2B Data Exchange properties of the DX_Event_Details transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Status</td>
<td>Status of the event.</td>
</tr>
<tr>
<td>Event Type</td>
<td>Type of the event.</td>
</tr>
<tr>
<td>Event Subject</td>
<td>Subject of the event. It is recommended to change the name to a unique value. Default value is <em>In Progress</em>.</td>
</tr>
<tr>
<td>Generate an error in case a failure occurs in this transformation</td>
<td>Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.</td>
</tr>
</tbody>
</table>

DX_Generate_Temporary_File Transformation

The DX_Generate_Temporary_File transformation generates a path and a file name for a document to store in the B2B Data Exchange document store.

Use the transformation to generate a unique file name in a temporary directory within the document store. For example, if you use an Unstructured Data transformation to write data to a temporary file, you can use the DX_Generate_Temporary_File transformation to generate a file name for a file in the document store. You then assign the file name and path to the file.

After you create the file in the temporary directory, you can use it in other transformations. For example, you can use the DX_Add_Document_To_Event transformation to attach the file to an event. PowerCenter copies the file from the temporary directory to the directory that contains all of the documents that it adds to events.

Input/Output Ports

Configure input/output ports for the DX_Generate_Temporary_File transformation on the Ports tab.

The following table describes the DX_Generate_Temporary_File input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXEventId</td>
<td>string</td>
<td>Required. Identifier of the event to associate with the generated file.</td>
</tr>
<tr>
<td>DXTemporaryFilePath</td>
<td>string</td>
<td>Path and file name of the temporary file.</td>
</tr>
<tr>
<td>DXErrorMessage</td>
<td>string</td>
<td>Error message generated by the transformation.</td>
</tr>
<tr>
<td>DXErrorCode</td>
<td>string</td>
<td>Error code generated by the transformation. If the transformation fails, the value of the DXErrorCode port is greater than zero.</td>
</tr>
</tbody>
</table>
B2B Data Exchange Properties

Configure properties for the DX_Generate_Temporary_File transformation on the DX Properties tab.

The following table describes the B2B Data Exchange properties of the DX_Generate_Temporary_File transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not use a temporary folder in the generated file path</td>
<td>Indicates whether to generate the file path within the temporary directory in the Data Exchange document store, or to generate the file path directly in the regular documents directory. Select this option to increase performance for documents that PowerCenter passes by reference and eliminate the additional backup that the temporary directory provides. <strong>Note:</strong> Select this option if you use the DX_Add_Document_To_Event transformation for an event with a document reference. Cleared by default.</td>
</tr>
<tr>
<td>Set the event status to Error if the transformation fails</td>
<td>Indicates whether to set the status of the event to Error when the transformation generates an error. Selected by default.</td>
</tr>
</tbody>
</table>

DX_Get_Document_File_Path Transformation

The DX_Get_Document_File_Path transformation gets the path and file name of a document from a document reference.

Input/Output Ports


The following table describes the DX_Get_Document_File_Path input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXEventId</td>
<td>string</td>
<td>Optional. Event associated with the document reference. If the value is not null, the transformation logs the event ID for tracking.</td>
</tr>
<tr>
<td>DXDocumentFilePath</td>
<td>string</td>
<td>Path and file name of the referenced document.</td>
</tr>
<tr>
<td>DXErrorMessage</td>
<td>string</td>
<td>Error message generated by the transformation.</td>
</tr>
<tr>
<td>DXErrorCode</td>
<td>string</td>
<td>Error code generated by the transformation. If the transformation fails, the value of the DXErrorCode port is greater than zero.</td>
</tr>
</tbody>
</table>
B2B Data Exchange Properties

Configure properties for the DX_Get_Document_File_Path transformation on the **DX Properties** tab.

The following table describes the B2B Data Exchange properties of the DX_Get_Document_File_Path transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate an error in case a failure occurs in this transformation</td>
<td>Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.</td>
</tr>
</tbody>
</table>

**DX_Increment_Profile_Parameter Transformation**

The DX_Increment_Profile_Parameter transformation increments the value of a parameter that is associated with a profile.

You can only increment numeric parameter values.

**Input/Output Ports**

Configure input/output ports for the DX_Increment_Profile_Parameter transformation on the **Ports** tab.

The following table describes the DX_Increment_Profile_Parameter input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXProfileId</td>
<td>string</td>
<td>Identifier of the profile that is associated with the parameter to increment.</td>
</tr>
<tr>
<td>DXValue</td>
<td>string</td>
<td>Value to use to increment the parameter. Default value is 1.</td>
</tr>
<tr>
<td>DXErrorMessage</td>
<td>string</td>
<td>Error message generated by the transformation.</td>
</tr>
<tr>
<td>DXErrorCode</td>
<td>string</td>
<td>Error code generated by the transformation. If the transformation fails, the value of the DXErrorCode port is greater than 0.</td>
</tr>
</tbody>
</table>
B2B Data Exchange Properties

Configure properties for the DX_Increment_Profile_Parameter transformation on the DX Properties tab.

The following table describes the B2B Data Exchange properties of the DX_Increment_Profile_Parameter transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile parameter name</td>
<td>Name of the profile parameter to increment.</td>
</tr>
<tr>
<td>Generate an error in case a failure occurs in this transformation</td>
<td>Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.</td>
</tr>
</tbody>
</table>

DX_Initiate_Correlation Transformation

The DX_Initiate_Correlation transformation initiates a reconciliation for an event. Use this transformation in a workflow that processes a document that requires reconciliation.

Every reconciliation that you initiate needs to be completed. After you create the workflow with the DX_Initiate_Correlation transformation, use the DX_Complete_Correlation in another workflow to complete the reconciliation.

Input Ports

Configure input ports for the DX_Initiate_Correlation transformation on the Ports tab.

The following table describes the DX_Initiate_Correlation input ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXCorrelationType</td>
<td>string</td>
<td>Required. Description of the correlation type to initiate. The description can contain up to 255 bytes. You configure the correlation type in the transformation properties.</td>
</tr>
<tr>
<td>DXCorrelationId</td>
<td>string</td>
<td>Required. Identifier of the correlation to initiate.</td>
</tr>
<tr>
<td>DXTimeInterval</td>
<td>integer</td>
<td>The time interval in seconds to complete the correlation.</td>
</tr>
</tbody>
</table>
Input/Output Ports

Configure input/output ports for the DX_Initiate_Correlation transformation on the Ports tab.

The following table describes the DX_Initiate_Correlation input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXEventId</td>
<td>string</td>
<td>Required. Identifier of the event that is associated with the correlation to initiate.</td>
</tr>
<tr>
<td>DXErrorMessage</td>
<td>string</td>
<td>Error message generated by the transformation.</td>
</tr>
<tr>
<td>DXErrorCode</td>
<td>string</td>
<td>Error code generated by the transformation. If the transformation fails, the value of the DXErrorCode port is greater than zero.</td>
</tr>
</tbody>
</table>

B2B Data Exchange Properties

Configure properties for the DX_Initiate_Correlation transformation on the DX Properties tab.

The following table describes the B2B Data Exchange properties of the DX_Initiate_Correlation transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate an error if the reconciliation ID already exists</td>
<td>Indicates whether to set the status of the event to error if the event is already associated with a correlation ID. Selected by default.</td>
</tr>
<tr>
<td>Generate an error in case a failure occurs in this transformation</td>
<td>Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.</td>
</tr>
</tbody>
</table>

DX_Profile_Parameters Transformation

The DX_Profile_Parameters transformation to get the parameter values defined for a profile.
Input/Output Ports

Configure input/output ports for the DX_Profile_Parameters transformation on the Ports tab.

The following table describes the DX_Profile_Parameters input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXProfileID</td>
<td>string</td>
<td>Required. Identifier of the profile.</td>
</tr>
<tr>
<td>DXEventId</td>
<td>string</td>
<td>Optional. Identifier of the associated event. You can use this port together with the ProfileId port.</td>
</tr>
<tr>
<td>DXErrorMessage</td>
<td>string</td>
<td>Error message that the transformation generates.</td>
</tr>
<tr>
<td>DXErrorCode</td>
<td>string</td>
<td>Error code that the transformation generates. If the transformation fails, the value of the DXError port is greater than zero.</td>
</tr>
</tbody>
</table>

The following table describes the DX_Profile_Parameters output-only ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXParametersXML</td>
<td>string</td>
<td>XML string that contains all the defined parameter names and values for the profile. Each required parameter must have a port. If the value represents a multiple selection, the value appears as a string that contains a semicolon-separated list of values. For example: &lt;parameter&gt;  &lt;parameter name=&quot;&quot; isnull={true/false}&quot; isdefault=&quot;true/false&quot;&gt;  ...CDATA value...  &lt;/parameter&gt;  To generate the XML string, the output port must be connected.</td>
</tr>
<tr>
<td>&lt;Parameter&gt;</td>
<td>string</td>
<td>Profile parameters that the transformation gets when you import a DX_Profile_Parameters transformation. The transformation populates the port from all the imported ports that are not default ports, such as eventid or profiled. Identifies the imported ports by name. Treats ports with the same name as a single parameter. If a parameter port does not match a defined parameter, B2B Data Exchange generates a debug log message. All &lt;parameter&gt; ports are strings. You can change the port type in the Operation Console with the Form Designer to one of the following types:  - Boolean. (true</td>
</tr>
</tbody>
</table>
**DX_Release_Delayed_Events Transformation**

The DX_Release_Delayed_Events transformation releases delayed events for a specified profile. The transformation releases the delayed events regardless of the delayed processing rule that you define for the profile.

**Input/Output Ports**

Configure input/output ports for the DX_Release_Delayed_Events transformation on the **Ports** tab. The following table describes the DX_Release_Delayed_Events input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXErrorMessage</td>
<td>string</td>
<td>Error message generated by the transformation.</td>
</tr>
<tr>
<td>DXErrorCode</td>
<td>string</td>
<td>Error code generated by the transformation. If the transformation fails, the value of the DXErrorCode port is greater than zero.</td>
</tr>
<tr>
<td>DXProfileId</td>
<td>string</td>
<td>Associated profile for the delayed processing rule. The profiles determines the events to release. Required if the value of the EventId list port is null.</td>
</tr>
<tr>
<td>EventId list</td>
<td>string</td>
<td>List of identifiers for the events to release, separated by commas. Required if the value of the DXProfileId port is null.</td>
</tr>
</tbody>
</table>

**B2B Data Exchange Properties**

Configure properties for the DX_Release_Delayed_Events transformation on the **DX Properties** tab. The following table describes the B2B Data Exchange properties of the DX_Release_Delayed_Events transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release events without checking the delayed processing rules</td>
<td>Indicates whether to release the event within the scope of the delayed processing rule. If true, the transformation releases the event even if it is not within the scope of any delayed processing rule. If false, the transformation releases the event based on the delayed processing rules. Cleared by default.</td>
</tr>
<tr>
<td>Generate an error in case a failure occurs in this transformation</td>
<td>Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.</td>
</tr>
</tbody>
</table>

**DX_Resolve_Profile**

The DX_Resolve_Profile transformation sets the profile ID based on the associated application and the partner or account number. The transformation returns an error if the combination of the application and the partner or account number is not associated with any profile.
Configure input ports for the DX.Resolve_Profile transformation on the Ports tab.

The following table describes the DX.Resolve_Profile input ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXApplication</td>
<td>string</td>
<td>Application that is associated with the profile.</td>
</tr>
<tr>
<td>DXEventId</td>
<td>string</td>
<td>Identifier of the associated event. If the transformation fails, the event status changes to error.</td>
</tr>
<tr>
<td>DXPartnerName</td>
<td>string</td>
<td>Required. Name of the partner that is associated with the profile.</td>
</tr>
<tr>
<td>DXAccountNumber</td>
<td>string</td>
<td>Account number that is associated with the profile. If you use the account number and partner name to resolve the profile, the account number is required. If you use the profile name to resolve the profile, the account number is not required.</td>
</tr>
</tbody>
</table>

Configure input/output ports for the DX.Resolve_Profile transformation on the Ports tab.

The following table describes the DX.Resolve_Profile input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXProfileId</td>
<td>string</td>
<td>Identifier of the profile that is associated with the application and with the partner or account number.</td>
</tr>
<tr>
<td>DXErrorMessage</td>
<td>string</td>
<td>Error message generated by the transformation.</td>
</tr>
<tr>
<td>DXErrorCode</td>
<td>string</td>
<td>Error code generated by the transformation. If the transformation fails, the value of the DXErrorCode port is greater than zero.</td>
</tr>
</tbody>
</table>

Configure properties for the DX.Resolve_Profile transformation on the DX Properties tab.

The following table describes the B2B Data Exchange properties of the DX.Resolve_Profile transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate an error in case a failure occurs in this transformation</td>
<td>Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.</td>
</tr>
</tbody>
</table>
The DX_Throw_Error transformation generates an error if an error occurs when the workflow runs.

Use the transformation to perform the following actions:

- Set the status of the associated event to error.
- Create the error message from the value of the DXDescription port.
- Attach the error message to the associated event.
- Send the error to the session log.

**Input Ports**

Configure input ports for the DX_Throw_Error transformation on the **Ports** tab.

The following table describes the DX_Throw_Error input ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXDescription</td>
<td>string</td>
<td>Description of the error to send to the session log. This error is also the description for the log document attached to the event.</td>
</tr>
<tr>
<td>DXMessageType</td>
<td>string</td>
<td>Optional. Type of the error event. Alphanumeric value to associate with the event. Any value is valid.</td>
</tr>
<tr>
<td>DXMIMEType</td>
<td>string</td>
<td>MIME type of the document to attach to the event.</td>
</tr>
</tbody>
</table>

**Input/Output Ports**

Configure input/output ports for the DX_Throw_Error transformation on the **Ports** tab.

The following table describes the DX_Throw_Error input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXEventId</td>
<td>string</td>
<td>Required. Identifier of the event that is associated with the error.</td>
</tr>
<tr>
<td>DXData</td>
<td>string</td>
<td>Log document to attach to the event. This port can contain the data of the document or a file path to the document. If the value of the parameter is null, the transformation creates an empty document and adds the document to the event. To attach a document with text data, set the datatype of the port to string or text. To attach a document with binary data, change the datatype of the port to binary.</td>
</tr>
</tbody>
</table>
| DXDataByReference | string | Indicates whether the DXData port contains the document data or a document reference. Possible values:  
- True. The DXData port contains a document reference.
- False. The DXData port contains the document data.
- Null. The DXData port contains the document data. |
<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXErrorMessage</td>
<td>string</td>
<td>Error message generated by the transformation.</td>
</tr>
<tr>
<td>DXErrorCode</td>
<td>string</td>
<td>Error code generated by the transformation. If the transformation fails, the value of the DXErrorCode port is greater than zero.</td>
</tr>
</tbody>
</table>

**B2B Data Exchange Properties**

Configure properties for the DX_Throw_Error transformation on the **DX Properties** tab.

The following table describes the B2B Data Exchange properties of the DX_Throw_Error transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error log document description</td>
<td>Description of the error log document that the transformation attaches to the event.</td>
</tr>
<tr>
<td>Message type</td>
<td>Alphanumeric value to associate with the event. You can enter any value.</td>
</tr>
<tr>
<td>Generate an error in case a failure occurs in this transformation</td>
<td>Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.</td>
</tr>
</tbody>
</table>
Chapter 3

Workflows

This chapter includes the following topics:

- Workflows Overview, 41
- Workflow Properties, 41

Workflows Overview

A workflow is a set of instructions that define how to process documents that partners send to B2B Data Exchange. A B2B Data Exchange workflow represents a PowerCenter workflow and mapping.

You can create a workflow in B2B Data Exchange from a PowerCenter workflow by either of these methods:

- Select a workflow directly from the PowerCenter repository.
- Export the PowerCenter workflow to an XML file using the Repository Manager tool. You then select the XML file to import as the workflow definition file.

Along with the workflow definition file, you define basic workflow properties, and save the new workflow.

You can create multiple workflows and associate them with an application. You can associate one workflow with one or more profiles. Real-time workflows use the JMS queue to communicate with PowerCenter.

If you assign event attributes that the dashboard uses in reports to a workflow, the event attribute value must be numeric. An event attribute is numeric if the lexical syntax rules in the Java Language Specification define it as a FloatValue.

Workflow Properties

A workflow in B2B Data Exchange contains properties such as name and description. You define the properties when you create or edit a workflow.

A workflow definition contains the following properties:

Workflow Name

Name of the workflow. The name can contain up to 60 characters and can include spaces and special characters.

Description

Description of the workflow. The description appears as a tooltip when the operator selects a workflow in the Profiles and Monitors pages.
Application

Name of the application to which the workflow belongs. Select <none> to create a workflow without an application.

Flow Type

The flow type depends on the application that is used to process documents.

- Select **PowerCenter Batch workflow** to associate the workflow with a PowerCenter batch workflow. A PowerCenter batch workflow reads from a file, database, or another source. The workflow runs once and stops after completion. You can use the PowerCenter batch workflows to process documents from B2B Data Exchange. You can also use the workflows to generate documents for B2B Data Exchange, such as outbound reports. Use the PowerCenter workflow export file as the workflow definition file.

- Select **PowerCenter Real-time workflow** to associate the workflow with a real-time workflow. A PowerCenter real-time workflow runs continuously and reads from real-time sources. You can use these workflows to process documents from B2B Data Exchange. Use the PowerCenter mapping or workflow export file as the workflow definition file.

- Select **Java (advanced)** to use Java classes as the flow engine in test mode or development mode. Use the Java class file as the workflow definition file.

PowerCenter Workflow

A PowerCenter workflow that you want to add to B2B Data Exchange. Choose one of the following options:

- **Select a workflow from the PowerCenter repository.** Select a PowerCenter workflow that you saved in the PowerCenter repository. To show all workflows in the repository, click Show All. To search for a workflow by name, enter a string in the Folder Name text box and then click Search.

- **Select a workflow definition file (XML).** Select an XML PowerCenter workflow definition file that you exported from the PowerCenter repository. To select a file, click the browse button, browse to the location of the required file, select the file, and then click Upload.
Creating Workflows for Event Reconciliation

This chapter includes the following topics:

- **Overview of Event Reconciliation**, 43
- **Initiating and Completing an Event Reconciliation**, 44
- **Reconciliation Status**, 45

Overview of Event Reconciliation

Reconciliation is the process of correlating an event with another event. For example, you send a document file to a partner containing transactions such as payments or orders that require acknowledgment. When you send the file to the partner, you initiate a reconciliation. When you receive an acknowledgment from the partner, you complete the reconciliation. B2B Data Exchange uses a correlation ID to identify each reconciliation.

One document can require multiple acknowledgments. For example, you send a partner an EDI document containing 10 purchase orders. Depending on the business requirements, you can require one acknowledgment for 10 purchase orders, or one acknowledgment for each of the 10 purchase orders. If you require one acknowledgment, you initiate one reconciliation. If you require 10 acknowledgments, you must initiate 10 reconciliations.

Use PowerCenter workflows to reconcile events. Event reconciliation typically involves two workflows. One workflow processes the event that requires reconciliation and initiates the reconciliation. The second workflow processes the acknowledgment received from the partner and completes the reconciliation. The reconciliation process uses the correlation ID to associate a document with the acknowledgment.

The reconciliation process involves the following steps:

1. B2B Data Exchange receives a document file that requires an acknowledgment and sends it to PowerCenter for processing.
2. The workflow designed to process the document initiates a reconciliation.
   - If the document requires multiple acknowledgments, the workflow initiates multiple reconciliations.
3. B2B Data Exchange sets the status for each reconciliation and sets the reconciliation status for the associated event.
5. The workflow designed to process the acknowledgment determines if the document completes a reconciliation.


### Initiating and Completing an Event Reconciliation

You can use the following transformations to initiate and complete reconciliations in PowerCenter:

- **DX_Initiate_Correlation.** Initiates a reconciliation for an event. When you create the workflow to process a document that requires reconciliation, use this transformation to initiate the reconciliation.

- **DX_Complete_Correlation.** Completes the reconciliation for an event. In the workflow to process the acknowledgment document, use this transformation to complete the reconciliation. This transformation requires the correlation ID defined by the workflow that initiates the reconciliation.

The B2B Data Exchange API also provides methods to initiate and complete a reconciliation. In the workflow to process a document that requires an acknowledgment, call the initiateCorrelation method to initiate a reconciliation. In the workflow to process the acknowledgment document, call the completeCorrelation method to complete the reconciliation.

You must set a time limit within which the acknowledgment must be received. The Data Exchange Server sets the reconciliation status to indicate whether the acknowledgment was processed within the time limit.

You can create a reconciliation monitor for the Data Exchange Server to set the status and to notify you when a reconciliation does not complete or completes after the time limit. For more information, see *B2B Data Exchange Operator Guide*.

### Initiating an Event Reconciliation

In the workflow to process a document that requires an acknowledgement, include the DX_Initiate_Correlation transformation to initiate a reconciliation. The DX_Initiate_Correlation transformation associates the reconciliation with an event ID and generates a correlation ID. It also sets the timeout window for the reconciliation to complete. For more information, see "DX_Initiate_Correlation Transformation" on page 34.

You can also call the initiateCorrelation method of the B2B Data Exchange API to initiate the reconciliation. The following example code shows how to call the initiateCorrelation method:

```java
String openingEventId = this.eventid;
String correlationId = doc.senderid + doc.receiverid + doc.transaction[i].id;
String correlationType = "397";
long timeWindowSeconds = 60*10; // 10 minutes
server.initiateCorrelation (openingEventId, correlationId, correlationType, timeWindowSeconds);
```

### Completing an Event Reconciliation

In the workflow that processes an acknowledgment document, include the DX_Complete_Correlation transformation to complete a reconciliation. The DX_Complete_Correlation transformation correlates the acknowledgment with the original document based on the correlation ID and completes the reconciliation. For more information, see "DX_Complete_Correlation Transformation" on page 26.
You can also call the completeCorrelation method of the B2B Data Exchange API to complete the reconciliation. The following example code shows how to call the completeCorrelation method:

```java
String closingEventId = this.eventId;
String correlationId = doc.receiveId + doc.senderId +
doc.acknowledgment[i].transactionId;
String openingEventId = server.completeCorrelation (closingEventId, correlationId);
```

Reconciliation Status

A reconciliation can have one of the following statuses:

- **Pending reconciliation.** No acknowledgment is received for a document and the timeout limit is not reached. When you initiate a reconciliation, the Data Exchange Server sets the reconciliation status to Pending reconciliation. When you change the timeout interval, the Data Exchange Server can reset the reconciliation status to Pending reconciliation.

- **Timed out.** Time limit for acknowledgment is reached and no acknowledgment is processed for the document.

- **Reconciled after timeout.** An acknowledgment for the document is processed after the time limit. When you complete a reconciliation after the time limit, the Data Exchange Server sets the reconciliation status to Reconciled after timeout.

- **Reconciliation complete.** An acknowledgment for the document is processed within the time limit. When you complete a reconciliation, the Data Exchange Server sets the reconciliation status to reconciliation complete. You can also manually set the reconciliation status to Reconciliation Complete on the Operation Console.
The following diagram shows the process that the Data Exchange Server follows to set the reconciliation status:

When an event has multiple reconciliations, the Data Exchange Server sets the reconciliation status of the event based on the status of all reconciliations associated with the event and its child events. The reconciliation status of the event is equivalent to the status of the reconciliation that is furthest from completion or requires operator action.

B2B Data Exchange ranks the reconciliation statuses in the following order:

1. Timed out. One or more of the reconciliations associated with the event and its child events have not received an acknowledgment within the time limit. This status can indicate a problem with the acknowledgment and require immediate operator action.
2. Pending reconciliation. All reconciliations associated with the event and its child events have been initiated but no reconciliation has completed or timed out.
3. Reconciled after timeout. All reconciliations associated with the event and its child events have completed but one or more of the reconciliations completed after the timeout. This status can indicate a problem with the acknowledgment process and require operator action.
4. Reconciliation complete. All reconciliations associated with the event and its child events are complete. This status indicates that the event reconciliation process worked as expected.
The reconciliation status of the event is equivalent to the highest ranked status of any of the reconciliations associated with the event and its child events. The following table shows examples of the event reconciliation status based on the status of the reconciliations associated with the event:

<table>
<thead>
<tr>
<th>Event</th>
<th>Status of the Reconciliations Associated with an Event</th>
<th>Reconciliation Status of the Event</th>
</tr>
</thead>
</table>
| Event 01 | Child event 101  
Reconciliation 001: Pending reconciliation  
Reconciliation 002: Reconciliation complete  
Reconciliation 003: Pending reconciliation  
Child event 102  
Reconciliation 011: Reconciliation complete  
Reconciliation 012: Reconciled after timeout  
Reconciliation 013: Reconciliation complete | Pending reconciliation |
| Event 02 | Child event 201  
Reconciliation 021: Reconciliation complete  
Reconciliation 022: Pending reconciliation  
Reconciliation 023: Reconciled after timeout  
Reconciliation 024: Pending reconciliation  
Child event 202  
Reconciliation 031: Timed out  
Reconciliation 032: Reconciled after timeout | Timed out |
| Event 03 | Reconciliation 041 Reconciliation complete  
Reconciliation 042: Reconciled after timeout | Reconciled after timeout |
Forms Designer Overview

The Forms Designer is a tool that developers and administrators can use to customize user interface and field behavior on Operations Console pages with a dynamic interface. You can use the Forms Designer to customize pages that display workflow parameters, partner attributes, and account attributes.

In the Forms Designer, you use the Designer tab to customize behavior and appearance of fields. For example, you can arrange large lists of workflow parameters in tabs or create dependencies between fields so that a specific parameter only appears if another parameter has a value. You can also mark fields as mandatory or read-only and determine the field type. For example, you can define a field as a text field or a number field. You can preview the customized layout on the Preview tab.

When you customize partner attributes, account attributes, or workflow parameters, you can import values from external lists with the B2B Data Exchange Forms Designer Java API. Use the API methods to configure the external list and set the default value for the list if needed. Each time you populate the list, the API overrides the values in the existing list. Therefore, if you manually modify any list value, the API deletes the value.

Forms Designer User Interface

The Designer tab displays the elements to customize, such as workflow parameters, partner attributes, or account attributes. You can use the Actions menu to add groups, such as tabs, columns, headers, or disclosures. You can also drag and drop elements to group and nest the fields in up to four levels.

The Designer tab displays information about the elements, such as label and type. The information appears in read-only mode.
The following table describes the element properties that you can view on the **Designer** tab:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Unique identifier of the element. For example, the name of the workflow parameter as defined in PowerCenter.</td>
</tr>
<tr>
<td>Label</td>
<td>Label of the element as it appears on the form. The label can be different from the ID of the element.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of the element, such as text or date.</td>
</tr>
<tr>
<td>Representation</td>
<td>Representation of the element in the form. For example, the element can be an entry field, a radio button, or a drop-down list.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Indicates whether the element is mandatory.</td>
</tr>
<tr>
<td>Read-Only</td>
<td>Indicates whether the element is read-only.</td>
</tr>
<tr>
<td>Depends On</td>
<td>Name of the element or group on which the parameter depends.</td>
</tr>
</tbody>
</table>

**Forms Designer Actions**

On the **Designer** tab, you can add a tab, column, header, or disclosure. If you select an element, you can edit the element properties.

The following table describes the actions that you can perform on the **Designer** tab:

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit</td>
<td>Opens a dialog box that you use to edit the properties of an element. Available when you select an element.</td>
</tr>
<tr>
<td>Create Tab</td>
<td>Creates a tab. You can drag and drop elements into tabs and move tabs up or down the element list to determine the tab order. The top tab is visible by default. You can nest tabs up to three levels.</td>
</tr>
<tr>
<td>Create Column</td>
<td>Creates a column. When you add two or more columns, elements that you drag and drop into a column appear side by side on the form. You can nest columns up to three levels.</td>
</tr>
<tr>
<td>Create Header</td>
<td>Creates a header. A header is a separator with a title that you use to group elements without hiding them from the form. You can nest headers up to three levels.</td>
</tr>
<tr>
<td>Create Disclosure</td>
<td>Create a disclosure. A disclosure is a header with a title and an expand arrow. When you click the arrow, an element group appears. By default, disclosures appear minimized on the form. You can nest disclosures up to three levels.</td>
</tr>
<tr>
<td>Delete Group</td>
<td>Deletes a tab, column, header, or disclosure. Deleting the group does not delete the elements from the form. Available when you select a group.</td>
</tr>
<tr>
<td>Move to Group</td>
<td>Moves elements from one tab, column, header, or disclosure group to another group. Available when you select an element.</td>
</tr>
</tbody>
</table>
Element Properties

When you edit an element in the Details dialog box, you can change element properties such as label, type, and default value. If you use the Forms Designer Java API, you can define an element to display values from external sources.

The dialog box appears when you select an element. If you select a group, you can edit only the label, type, and field dependency for the group.

The following table describes the element properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Required. Name of the element to display in the form. The label can be different from the ID.</td>
</tr>
<tr>
<td>Description</td>
<td>Textual description of the parameter. The description appears in a tooltip when you hover over the element in the form.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Requires users to enter a value for the element in the form.</td>
</tr>
<tr>
<td>Read Only</td>
<td>Defines the element as read-only and the users cannot change the element value.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Hides the element from the form. Select this check box for workflow parameters that operators cannot edit.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of data for the element value. Choose one of the following options:</td>
</tr>
<tr>
<td></td>
<td>- Boolean. If you select this option, the Check box representation is selected by default.</td>
</tr>
<tr>
<td></td>
<td>- Date. If you select this option, the Date picker representation is selected by default.</td>
</tr>
<tr>
<td></td>
<td>- Number. If you select this option, the Entry field representation is selected by default.</td>
</tr>
<tr>
<td></td>
<td>- Text</td>
</tr>
<tr>
<td>Representation</td>
<td>Type of input that users enter for the element value. Choose one of the following options:</td>
</tr>
<tr>
<td></td>
<td>- Entry field</td>
</tr>
<tr>
<td></td>
<td>- Dropdown list</td>
</tr>
<tr>
<td></td>
<td>- Radio button</td>
</tr>
<tr>
<td></td>
<td>- List</td>
</tr>
<tr>
<td></td>
<td>- Check box</td>
</tr>
<tr>
<td>Default Value</td>
<td>Value that appears when the form appears. If you select a list or dropdown list representation and populate the list with the Forms Designer Java API, the external data source determines the default value.</td>
</tr>
<tr>
<td>Values</td>
<td>A list of valid values for the parameter. If you select a list or dropdown list representation, enter one or more values. The values list is not case sensitive. <strong>Note:</strong> If you populate list values from an external source, do not change the values in the list. The values from the external source override any manual changes you make to the list.</td>
</tr>
<tr>
<td>Minimum</td>
<td>Minimum valid value for the element. For string elements, the value you enter in this property determines the minimum number of characters. For number elements, the value you enter in this property determines the minimum numeric value. Available when you select a string or number element type.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maximum</td>
<td>Maximum valid value for the element. For string elements, the value you enter in this property determines the maximum number of characters. For number elements, the value you enter in this property determines the maximum numeric value. Available when you select a string or number element type.</td>
</tr>
<tr>
<td>Field Dependency</td>
<td>Set of conditions that determines whether to link the element to another element or group in the form. For example, you can choose to enable the element only after users enters a value in a different element.</td>
</tr>
</tbody>
</table>

### Group Properties

When you create or edit a tab, column, header, or disclosure in the **Details** dialog box, you can change the label, type or field dependency properties.

The following table describes the group properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Mandatory. Name of the group to display in the form.</td>
</tr>
<tr>
<td>Type</td>
<td>Mandatory. Type of the group. Choose from one of the following options:</td>
</tr>
<tr>
<td></td>
<td>- Tab</td>
</tr>
<tr>
<td></td>
<td>- Column</td>
</tr>
<tr>
<td></td>
<td>- Header</td>
</tr>
<tr>
<td></td>
<td>- Disclosure</td>
</tr>
<tr>
<td>Field Dependency</td>
<td>Set of conditions that determines whether to link the group to another element or group in the form. For example, you can choose to enable a tab only after users entered a specific value in a different element.</td>
</tr>
</tbody>
</table>

### Customizing Workflow Parameters

Use the Forms Designer to customize user interface layout and field behavior for workflow parameters in the Operation Console and in the Partners Portal Message Profile wizard. The steps in this task are similar to the steps you follow to customize partner attributes and account attributes.

Before you customize workflow parameters, create or edit a workflow in the Forms Designer. Ensure that the PowerCenter workflow contains workflow parameters.

1. In the Navigator, click **Partner Management > Workflows**.
2. Choose to create or edit a workflow.
   - To create a workflow, import that workflow XML file from PowerCenter. Ensure that the workflow you import contains workflow, mapping, or session parameters.
   - To edit a workflow, click **Edit** next to the workflow that you want to edit.
3. Click the **Workflow Parameters** tab.
The Preview pane displays the default view of the workflow parameter list. The Portal Parameters tab displays the list of workflow parameters that are shown and can be edited in the Message Profile wizard in the Partners Portal.

**Note:** The name of the Portal Parameters tab is configurable, and the tab might have another name.

4. Click Designer.

The Designer pane displays the parameter list in alphabetical order.

5. To arrange the parameters and groups, drag the element or group you want to arrange.

To create a list of workflow parameters to display for message profiles in the Partners Portal, drag the element or group you want to arrange under the Portal Parameters tab.

6. To change the behavior and appearance of a parameter element, click the parameter and click **Actions > Edit**.

The Details dialog box appears.

7. Change the properties for the parameter and click **Save**.

**Note:** The Label property is a required property.

8. To add a group, click the Actions menu and choose to add a tab, column, header, or disclosure.

9. To preview the customized form, click **Preview** and verify the appearance of the form and any field dependencies you defined.

10. Click **Save** to save the workflow.

The customized form appears when you create or edit a profile and associate the workflow with the profile.
Overview of Email for Monitor Notification

B2B Data Exchange generates events as it processes documents. Managing objects in the Operation Console also creates events, called logging events.

Operators can track events by using event monitors. An event monitor can be created to track events for different profiles or to track different types of events. The creator of an event monitor can specify email notification when the monitor detects an event. For more information about event monitors, see the B2B Data Exchange Operator Guide.

B2B Data Exchange provides a built-in email notification message format. The developer can modify this format by setting certain system and event properties. The user can provide his own email message format.

The Built-in Email Notification Message Format

The built-in format of the email notification message that monitors send contains the following information:

- General information. The message includes details such as the monitor that sent the message, partner and account details, and a link to the event. If you are logged on to the B2B Data Exchange console, clicking on the link displays event details.
- Log details. The message includes a list of all log entries for this event, including document attachments.
- Reconciliation details. Acknowledgment details received from the partner for this event.
Configuring the Email Notifications

You can use system properties and event properties to customize the email properties.

You can completely customize the body of the email message.

Configuring Email Notifications Using System Properties

Use the dx_email_from_field system property to customize the email notification that monitors send.

For more information, see the B2B Data Exchange Administrator Guide.

Configuring Email Notifications Using Event Properties

The following table defines the event properties that you can use to modify built-in email notification messages. Use the DX_EVENT_ATTRIBUTE transformation to set the attributes for an event.

<table>
<thead>
<tr>
<th>Event Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dx_email_bcc_field</td>
<td>A semicolon-separated list of recipient email addresses. These recipients are added to the addressees defined in the monitor.</td>
</tr>
<tr>
<td>dx_email_body_file</td>
<td>The path to the file containing the custom body of the email. This file replaces the text of the body of the built-in email.</td>
</tr>
<tr>
<td>dx_email_body_field</td>
<td>A character string that is used to replace the body text of the built-in email. Maximum length is 255 characters.</td>
</tr>
<tr>
<td>dx_email_from_field</td>
<td>This string becomes the &quot;from field&quot; of the email.</td>
</tr>
</tbody>
</table>
### Event Property Description

<table>
<thead>
<tr>
<th>Event Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dx_email_cc_field</td>
<td>A semicolon-separated list of recipient email addresses. These recipients are added to the addressees defined in the monitor.</td>
</tr>
<tr>
<td>dx_email_mimetype</td>
<td>Default is &quot;text/html&quot;; charset=UTF-8. For more information about mime types, see <a href="http://www.mimetype.org">http://www.mimetype.org</a>.</td>
</tr>
<tr>
<td>dx_email_subject_field</td>
<td>Subject field of the email.</td>
</tr>
<tr>
<td>dx_email_to_field</td>
<td>A semicolon-separated list of recipient email addresses. These recipients are added to the list of addressees defined in the monitor.</td>
</tr>
</tbody>
</table>

### Configuring the Body of an Email Notification

There are two ways to customize the body of an email notification:

- **Provide a text or HTML file that contains the body of the customized email.** Assign the path of this file to the dx_email_body_file event property. This file must be accessible to all Data Exchange Server instances.

- **Provide a text string that will replace the built-in body text.** Assign this string to the dx_email_text event property.

The customized body of an email message can contain variables that are set during runtime.

The following variables are built-in:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$DXMonitorName</td>
<td>The name of the monitor.</td>
</tr>
<tr>
<td>$DXPartnerInfo</td>
<td>Partner name, account name, and account number.</td>
</tr>
</tbody>
</table>

You can use event attributes in the customized body of an email. For example, you can define an event attribute called MyAttribute and embed it in the body of an email as $MyAttribute.

The developer assigns a value to MyAttribute using the DX_Event_Attribute transformation.
Chapter 7

Dashboard and Reports

This chapter includes the following topics:

- Dashboard and Reports Overview, 56
- Key Performance Indicators, 57
- Dashboard and Reports Structure in Logi Info Studio, 58
- Customizing the Dashboard in Logi Info Studio, 62

Dashboard and Reports Overview

The Dashboard is a collection of panels that contain reports about information that B2B Data Exchange processes. Use the Dashboard to view personalized visual summary information about B2B Data Exchange event and document processing, such as the number of events for certain partners or the error rate for specific accounts.

Most of the reports in the Dashboard are based on key performance indicators (KPIs) that B2B Data Exchange retrieves from the operational data store, such as events, errors, transactions, and values. The operational data store is a repository that contains aggregated information solely for reporting purposes. The Dashboard retrieves the aggregated events from the operational data store and displays the information in the panels. In addition to the default KPIs, you can select event attributes in the Operation Console to load to the operational data store as custom KPIs.

You use Logi Info Studio to create custom reports based on default KPIs or custom KPIs. In Logi Info Studio, you set up the custom Dashboard and create custom Dashboard panels that include KPIs based on the information that you want to display. You deploy the custom Dashboard in B2B Data Exchange and modify the connection properties to retrieve the custom Dashboard panels that you created.


The software available for download at the referenced links belongs to a third party or third parties, not Informatica LLC. The download links are subject to the possibility of errors, omissions or change. Informatica assumes no responsibility for such links and/or such software, disclaims all warranties, either express or implied, including but not limited to, implied warranties of merchantability, fitness for a particular purpose, title and non-infringement, and disclaims all liability relating thereto.

The B2B Data Exchange administrator installs the operational data store with the main B2B Data Exchange installation and modifies B2B Data Exchange system properties to determine certain aspects of the Dashboard behavior, such as whether to show the Dashboard when users log on to the Operation Console and the error rate range in the Error Rate Gauge panel. Operators can personalize the Dashboard page to display different panels and apply filters based on the information that the operators want to view.
After the B2B Data Exchange administrator installs the Dashboard and Reports component, they use PowerCenter Workflow Manager to configure the operational data store event loader. The operational data store event loader is a PowerCenter workflow that collects KPIs from the Data Exchange repository according to specified parameters and loads aggregated events to the operational data store. In the operational data store event loader, the B2B Data Exchange administrator defines parameters such as the number of retry attempts for each event load process in case of failure or the number of minutes to wait between event load processes.

### Key Performance Indicators

Use key performance indicators (KPIs) to collect information about events that B2B Data Exchange generates. The operational data store event loader collects KPIs from the main B2B Data Exchange repository and transfers aggregated KPIs to the Dashboard.

The Dashboard displays most of the reports based on default KPIs that B2B Data Exchange retrieves from the operational data store. In addition to the default KPIs, you can select event attributes to load to the operational data store as custom KPIs. You use Logi Info Studio to create custom Dashboard panels based on the custom KPIs that you select. Use numeric event attributes in custom Dashboard panels.

In addition to reports from the operational data store, B2B Data Exchange retrieves unresolved error event information directly from the runtime B2B Data Exchange. To maintain performance, do not customize the unresolved error event panels or create custom Dashboard panels based on event information from the runtime B2B Data Exchange repository.

If you track service level agreement (SLA) requirements, you can create rules based on default KPIs or custom KPIs and view violations in the SLA Violations panel of the Dashboard.

### Default KPIs

The operational data store uses default KPIs in reports. The operational data store event loader collects the KPIs and loads the KPIs to the operational data store.

Operation Console users can view the aggregated event information as reports in the Dashboard panels. You can create custom Dashboard panels in Logi Info Studio based on the default KPIs. When you create custom Dashboard panels, you can apply measurements on the default KPIs, such as sum, count, minimum, maximum, and average.

The following table describes the default KPIs that the Dashboard displays:

<table>
<thead>
<tr>
<th>KPI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message processing time</td>
<td>Duration in minutes of the time it takes for the event to initially reach a final state.</td>
</tr>
<tr>
<td>Number of events</td>
<td>Number of events that B2B Data Exchange processes.</td>
</tr>
<tr>
<td>Number of error events</td>
<td>Number of error events that reached a final state.</td>
</tr>
</tbody>
</table>
Dashboard and Reports Structure in Logi Info Studio

The Dashboard displays event information based on aggregated data that it retrieves from the operational data store. Use Logi Info Studio to view the default Dashboard panels and create custom Dashboard panels based on default KPIs and custom KPIs that you retrieve from the operational data store.

The default Dashboard in Logi Info Studio includes the following global elements:

- Filters that the Dashboard can apply to each report.
- Theme modifier that controls the appearance of the Dashboard panels.
- SQL constants for common queries to run in the operational data store.
- JavaScript support files that B2B Data Exchange uses to process reports and display Dashboard panels.

Default Dashboard Filters

The Dashboard uses filters for each report that you generate in Logi Info Studio. The filters correspond the global filters in the Datasboard page of the Operation Console.

The following table describes the default Dashboard filters:

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@Request.idBegin~</td>
<td>Start date of the date/time filter, in the format yyyy-mm-dd.</td>
</tr>
<tr>
<td>@Request.itBegin~</td>
<td>Start time of the date/time filter, in the format hh:mm.</td>
</tr>
<tr>
<td>@Request.idEnd~</td>
<td>End date of the date/time filter, in the format yyyy-mm-dd.</td>
</tr>
<tr>
<td>@Request.itEnd~</td>
<td>End time of the date/time filter, in the format hh:mm.</td>
</tr>
<tr>
<td>@Request.iPartnerId~</td>
<td>Partner database identifier for the selected partner.</td>
</tr>
<tr>
<td>@Request.iAccountId~</td>
<td>Account database identifier of the selected account.</td>
</tr>
<tr>
<td>@Request.eventTypeId~</td>
<td>Event type database identifier of the selected event type.</td>
</tr>
<tr>
<td>@Request.eventStatusId~</td>
<td>Event status database identifier of the selected event status.</td>
</tr>
</tbody>
</table>

Default Dashboard SQL Constants

Each report in the default Dashboard uses SQL constants for common queries to the operational data store and query parameters. You can reuse these SQL constants when you create custom Dashboard panels based on KPIs that you load to the operational data store.

**Note:** For the SQL server, use the same constants with the addition of the suffix _SQLSERVER. For example, DX_ODS_FILTER_ACCOUNT would be DX_ODS_FILTER_ACCOUNT_SQLSERVER.

**Note:** You cannot create reports in Logi Info Studio with SQL queries to the run-time B2B Data Exchange repository.
The following table describes the default Dashboard SQL constants:

<table>
<thead>
<tr>
<th>SQL Constant Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DX_ODS_FILTER_ACCOUNT</td>
<td>Filters the database rows according to the related account. Includes the following query parameters:</td>
</tr>
<tr>
<td></td>
<td>- requestAccountId</td>
</tr>
<tr>
<td>DX_ODS_FILTER_ERROR_STATE</td>
<td>Filters the database rows to display only error events.</td>
</tr>
<tr>
<td>DX_ODS_FILTER_EVENT_STATUS</td>
<td>Filters the database rows according to the event status. Includes the following query parameters:</td>
</tr>
<tr>
<td></td>
<td>- requestEventStatusId</td>
</tr>
<tr>
<td>DX_ODS_FILTER_EVENT_STATUS_STATE</td>
<td>Filters the database rows according to the event state. The following values are available:</td>
</tr>
<tr>
<td></td>
<td>- 1. Show error events with a final state.</td>
</tr>
<tr>
<td></td>
<td>- 2. Show non-error events with a final state.</td>
</tr>
<tr>
<td></td>
<td>- -1. Show all events with a final state.</td>
</tr>
<tr>
<td></td>
<td>Includes the following query parameters:</td>
</tr>
<tr>
<td></td>
<td>- requestEventStatusState</td>
</tr>
<tr>
<td>DX_ODS_FILTER_EVENT_TYPE</td>
<td>Filters the database rows according to the event type. Includes the following query parameters:</td>
</tr>
<tr>
<td></td>
<td>- requestEventTypeId</td>
</tr>
<tr>
<td>DX_ODS_FILTER_PARTNER</td>
<td>Filters the database rows according to the related partner. Includes the following query parameters:</td>
</tr>
<tr>
<td></td>
<td>- requestPartnerId</td>
</tr>
<tr>
<td>DX_ODS_FILTER_TIME_ORACLE</td>
<td>Filters the database rows according to a selected time frame. Includes the following query parameters:</td>
</tr>
<tr>
<td></td>
<td>- requestDateBegin</td>
</tr>
<tr>
<td></td>
<td>- requestTimeBegin</td>
</tr>
<tr>
<td></td>
<td>- requestDateEnd</td>
</tr>
<tr>
<td></td>
<td>- requestTimeEnd</td>
</tr>
<tr>
<td>DX_ODS_USER_RESTRICTION_BY_ACCESS_ID</td>
<td>Filters the database rows according to the related user access identifier. Note: To maximize user access control, use DX_ODS_USER_RESTRICTION_BY_ACCESS_ID, DX_ODS_USER_RESTRICTION_BY_EVENT_TYPE, and DX_ODS_USER_RESTRICTION_BY_PARTNER in your custom Dashboard panels. Includes the following query parameters:</td>
</tr>
<tr>
<td></td>
<td>- requestUserId</td>
</tr>
</tbody>
</table>
### SQL Constant Name

<table>
<thead>
<tr>
<th>SQL Constant Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| DX_ODS_USER_RESTRICTION_BY_EVENT_TYPE                  | Filters the database rows to display only event types for which the user has viewing privileges.  
**Note:** To maximize user access control, use DX_ODS_USER_RESTRICTION_BY_ACCESS_ID, DX_ODS_USER_RESTRICTION_BY_EVENT_TYPE, and DX_ODS_USER_RESTRICTION_BY_PARTNER in your custom Dashboard panels.  
Includes the following query parameters:  
- requestUserId                                                                                                                                                                                                 |
| DX_ODS_USER_RESTRICTION_BY_PARTNER                     | Filters the database rows to display only related partners for which the user has viewing privileges.  
**Note:** To maximize user access control, use DX_ODS_USER_RESTRICTION_BY_ACCESS_ID, DX_ODS_USER_RESTRICTION_BY_EVENT_TYPE, and DX_ODS_USER_RESTRICTION_BY_PARTNER in your custom Dashboard panels.  
Includes the following query parameters:  
- requestUserId                                                                                                                                                                                                 |

### Default Dashboard JavaScript Support Files

B2B Data Exchange uses JavaScript files and jQuery files to process event information when you view the Dashboard in the Operation Console. The files are located in `<DXInstallDir>\LogiXML\DXDashboard\_SupportFiles`.

The following table describes the JavaScript files and jQuery files that the default Dashboard uses:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dx.js</td>
<td>Functions that the Data Exchange server uses when it processes reports with the LogiXML engine.</td>
</tr>
<tr>
<td>dxbrowser.js</td>
<td>Functions that the browser client uses when you view the Dashboard in the Operation Console.</td>
</tr>
<tr>
<td>jquery-<em>.</em></td>
<td>Collection of jQuery files that enable dynamically-populating drop down lists in the global filter section, such as partner or account.</td>
</tr>
</tbody>
</table>
Default Dashboard Theme Modifier

The theme modifier controls the appearance of the Dashboard panels. B2B Data Exchange applies the style and appearance settings in the theme modifier to any custom Dashboard panel that you create. The theme modifier files are located in `<DXInstallDir>\LogiXML\Themes\DXDashboard\Informatica`.

The following table describes the theme modifier components for the default dashboard:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theme.css</td>
<td>Cascading style sheet that determines the graphical and textual formatting of the Dashboard panels.</td>
</tr>
<tr>
<td>ThemeModifier.xml</td>
<td>Global settings for the Dashboard panels behavior and the single sign-on mechanism that B2B Data Exchange uses when you log on to the Operation Console.</td>
</tr>
</tbody>
</table>

Custom Dashboard Properties in the LogiXML Settings File

You modify certain properties in the LogiXML _Settings.lgx file when you create a custom Dashboard for B2B Data Exchange.

The following table describes the properties to modify:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DX_CONSOLE_URL</td>
<td>Location of the Operation Console, in the following format: http://&lt;dx host&gt;;&lt;dx port&gt;/dx-console</td>
</tr>
<tr>
<td>DASHBOARD_SAVEFOLDER</td>
<td>Location to store the personalized layout settings for each Dashboard user. Default is: @Function.AppPhysicalPath-\dx\saved_dashboards</td>
</tr>
<tr>
<td>DX_ODS_ORACLE_CONNECTION</td>
<td>Connection string for the operational data store.</td>
</tr>
<tr>
<td>DX_ORACLE_CONNECTION</td>
<td>Connection string for the main B2B Data Exchange repository.</td>
</tr>
<tr>
<td>LogonFailPage</td>
<td>URL to display after the user logs out of the Operation Console. The default URL is the B2B Data Exchange Logout page: http://&lt;dx host&gt;:&lt;dx port&gt;/dx-console/logout.jsp</td>
</tr>
<tr>
<td>AuthenticationClientAddresses</td>
<td>IP addresses for servers that can request authentication tokens from the LogiXML server, separated by semicolons. For each server, set IP4 and IP6 addresses.</td>
</tr>
<tr>
<td>caption</td>
<td>Description of the custom Dashboard.</td>
</tr>
<tr>
<td>COOKIE_PATH</td>
<td>Path to the cookie file for the custom Dashboard. Must match the value in the Cookie Path property.</td>
</tr>
<tr>
<td>WEBAPP_NAME</td>
<td>Name of the folder which contains the custom Dashboard. Default is: dxdashboard</td>
</tr>
</tbody>
</table>

Redirect Error URL | Link for the page to display if the Dashboard encounters an error. |
Customizing the Dashboard in Logi Info Studio

Use Logi Info Studio to create custom Dashboard panels based on default KPIs or custom KPIs.

In Logi Info Studio, you create a report and add SQL queries that retrieve numeric event attributes from the operational data store. You then configure a Dashboard panel to display the custom report and deploy the custom Dashboard in B2B Data Exchange.

You cannot create custom Dashboard panels based on the following reports:

- Unresolved Error Events by Partner (eventsbypartnerErrorRepository.lgx)
- Unresolved Error Events by Account (eventsbyaccountErrorRepository.lgx)
- Unresolved Error Events by Event Type (eventsbyeventstatusErrorRepository.lgx)
- Unresolved Error Events by Event Status (eventsbyeventstatusErrorRepository.lgx)
- Tasks (dxtasks.lgx)

When you create custom reports in Logi Info Studio, you can copy the SQL constants from the default Dashboard and modify them based on the KPIs that you want to display in the Dashboard panels.

You can use the SQL constants with the LogiXML constant token. When Logi Info Studio processes the custom reports, the constant token expands the SQL constants. For example, you use the following SQL query:

```
select facts.ACCOUNT_ID, sum(EVENT_COUNT) as SUM_COUNT from DX_ODS_EVENT_FACTS facts
where @Constant.DX_ODS_FILTER_TIME_ORACLE-
```

The LogiXML constant token expands the SQL query to the following SQL query:

```
select facts.ACCOUNT_ID, sum(EVENT_COUNT) as SUM_COUNT from DX_ODS_EVENT_FACTS facts
where to_timestamp(:requestDateBegin || :requestTimeBegin,'YYYY-MM-DD HH24:MI:SS') <=
    facts.TIMESLICE
and facts.TIMESLICE <= to_timestamp(:requestDateEnd || :requestTimeEnd,'YYYY-MM-DD
HH24:MI:SS')
```

If you create new parameters for your custom report, you must specify a unique name for each parameter. Otherwise, the Dashboard overrides the existing parameters with the same name or processes the parameter values incorrectly. If you create data layer links, you must specify a unique name for each data layer link in all of the reports. Otherwise, the Dashboard may display incorrect data in the Dashboard panel that you create, such as the wrong date or status.

Installing and Registering Logi Info Studio

You install Logi Info Studio to enhance and create Dashboard panels for B2B Data Exchange.


1. In the Logi Info Studio installer, choose a custom installation and select to install only the Studio component.
2. Contact Informatica shipping to receive the Logi Info Studio license file.
3. Copy the Logi Info Studio license file to the following folder: `<LogiXMLInstallationDir>\LogiStudio`
Setting Up the Custom Dashboard Application

You create the custom Dashboard as an application in Logi Info Studio and import the default Dashboard files to Logi Info Studio.

1. In Logi Info Studio, create a Java application.
2. In the Prepare New Application wizard, configure the name and location for the Dashboard application:
   • In the application name, enter the name for the custom Dashboard.
   • In the deployment folder location, enter `<DXInstallDir>\DataExchange\tomcat_server_version> \webapps.
3. Skip the remaining Prepare a New Application wizard screens.
4. Copy the contents of the `<DXInstallDir>\LogiXML\dxdashboard folder to the newly created `<DXInstallDir>\DataExchange\tomcat_server_version> \webapps\<custom_dashboard> folder.
5. Copy the LogiXML license file from the `<DXInstallDir>\DataExchange\tomcat_server_version> \webapps\dxdashboard folder to the `<DXInstallDir>\DataExchange\tomcat_server_version> \webapps\<custom_dashboard> folder.
6. In the LogiXML _Settings.lgx file, modify the properties according to the name and location of the custom Dashboard.
7. In the B2B Data Exchange server, replace the value of the dx.dashboard.url system property with the value of the URL for the custom dashboard in the following format: http://<host>:<port>/ <custom_dashboard>

Creating a Report in the Custom Dashboard

After you set up the custom Dashboard in Logi Info Studio, you create and configure a report to use in a custom Dashboard panel. You can copy an existing report and modify it as needed.

1. In Logi Info Studio, create a report definition.
2. Select the report filters that you want to provide to the users.
3. Define the SQL parameters for the SQL query to run in the operational data store. You must define the name and type of each parameter.

Adding a Dashboard Panel to the Custom Dashboard

After you create a report in Logi Info Studio, you add the custom Dashboard panel and its elements to the main Dashboard report file.

1. In Logi Info Studio, open the custom Dashboard report file.
2. In the dxdashboard element, add a new Panel element with a caption and a unique identifier.
3. In the Panel element, add a Panel Content element.
4. In the Panel Content element, add a SubReport element with a unique identifier and set the property Frame Border to false.
5. In the SubReport element, add a TargetReport element with a unique identifier and enter the name of the new report in the Report Definition File property.
6. Copy the Link Parameters element from an existing SubReport element to the new SubReport element.
7. Save the dxdashboard report file.
Deploying and Testing the Custom Dashboard in B2B Data Exchange

After you modify or create a custom Dashboard in Logi Info Studio, you copy the custom Dashboard application folder contents to B2B Data Exchange. You then test that the custom Dashboard appears correctly.

1. Copy the entire custom Dashboard application folder to the following location: \DXInstallDir\DataExchange\<tomcat_server_version>\webapps.

2. Back up the _Setting.lgx file and modify the following settings:
   - Delete the value of the Redirect Error URL property to allow access to the LogixML error report.
   - Set the value of the Debugger Style property to DebuggerLinks to add links to the LogixML progress report. The Dashboard panel layout may display differently.

3. Log on to the Operation Console and test that the new Dashboard panels open and populate correctly.

4. After you test the custom Dashboard, restore the properties in the _Setting.lgx file.
Data Exchange Web Services Overview

B2B Data Exchange provides a set of Web services that you can use to create and manage partners, accounts, and profiles from a client application. You can also use the Web services to send input documents to B2B Data Exchange.

When you install B2B Data Exchange, the Data Exchange Web services are installed as a set of XML files. The XML files are PowerCenter export files that contain Web service workflows. To use the Web services, import the XML files into PowerCenter and run the Web service workflows. Then you can call the Web service operations from any Web service client application.

B2B Data Exchange provides the following Web services:

- **DX_TPM_Partner**: Creates and manages trading partners.
- **DX_TPM_Account**: Creates and manages accounts.
- **DX_TPM_Profile**: Creates and manages profiles.
- **DX_Endpoint**: Sends documents to B2B Data Exchange.
DX_TPM_Partner Web Service

The DX_TPM_Partner web service creates, deletes, and manages trading partners in the B2B Data Exchange repository.

Use the DX_TPM_Partner web service to perform the following tasks:

- Create a partner.
- Delete a partner.
- Update the information for a partner.
- Search for a partner.
- Count of the number of partners in the repository.
- Get information for a partner by index.

Reserved System Partner

Do not use the name none when you create, update, or search for a partner. This name is reserved for a dedicated system partner. The web service returns a fault response if you try to assign this name to another partner.

DX_TPM_Partner Request

The DX_TPM_Partner request defines the task performed by the DX_TPM_Partner web service.

The DX_TPM_Partner request message includes the following elements:

**operation**

Required. Type of operation the web service will perform. The web service can perform one of the following operations:

- create
- delete
- update
- search
- count
- get

**partner**

Complex type object DXPartner that includes elements that describe the trading partner to create. Required by the create and update operations and ignored by other operations.

**account**

Complex type object DXAccount that includes elements that describe the partner account to create or update. Required by the create operation and ignored by other operations.

**partnerName**

Name of the partner to search or delete. Required by the search and delete operations and ignored by other operations.

**index**

Index for the partner to get. The index number starts at zero. Required by the get operation and ignored by other operations.
DX_TPM_Partner Response

The DX_TPM_Partner response is the reply from the DX_TPM_Partner web service.

The DX_TPM_Partner response message includes the following elements:

- **partner**
  Complex type object DXPartner that includes elements that describe the trading partner to create or update. The partner element is included in the response message for the following operations:
  - create
  - update
  - get
  - search

- **count**
  Number of partners in the B2B Data Exchange repository. The count element is included in the response message for the count operation.

**Note:** The DX_TPM_Partner web service sends an empty response to the client when the web service requests a delete operation.

DX_TPM_Partner Create Operation

The DX_TPM_Partner create operation creates a partner and an associated account.

**Guidelines**

Use the following guidelines when you create a partner:

- The request must use the create operation.

**Response**

The web service returns a DXPartnerResponse object that includes the partner information in a DXPartner object.

The web service returns a fault response in the following situations:

- The DXPartner object has an element with an invalid value.
- The DXAccount object has an element with an invalid value.
- A partner in the Data Exchange repository has the same partner name specified in the DXPartner object.
- The name is already reserved for a dedicated system partner.

**Create SOAP Message Example**

The following SOAP message shows an example of a request to the DX_TPM_Partner web service to create a partner:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tpm="http://b2b.informatica.com/dx/tpm/">
  <soapenv:Header>
    <inf:Security>
      <UsernameToken>
        <Username>Administrator</Username>
        <Password>Administrator</Password>
      </UsernameToken>
    </inf:Security>
  <sec:DXSecurity>
    <sec:DXUsernameToken>
```
<?xml version="1.0" encoding="UTF-8"?>
                     xmlns:tns="http://schemas.xmlsoap.org/soap/envelope/"
                     xmlns:ns="urn:schemas-netscape悒OTA憧憬API"
                     xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <tns:DXSecurity/>
  </soapenv:Header>
  <soapenv:Body>
    <tns:DXPartnerRequest>
      <operation>create</operation>
      <partner>
        <Name>PartnerTestName</Name>
        <Description>PartnerTest Description</Description>
        <ContactInformation>
          <Name>PartnerTestContact1</Name>
          <Address>PartnerTestContact1 Address</Address>
          <PhoneNumber>+12345678912</PhoneNumber>
          <FaxNumber>+12345678913</FaxNumber>
          <EmailAddress>PartnerTestName1@informatica.com</EmailAddress>
        </ContactInformation>
        <ContactInformation>
          <Name>PartnerTestContact2</Name>
          <Address>PartnerTestContact2 Address</Address>
          <PhoneNumber>+12345678922</PhoneNumber>
          <FaxNumber>+12345678923</FaxNumber>
          <EmailAddress>PartnerTestName2@informatica.com</EmailAddress>
        </ContactInformation>
        <PartnerAttribute>
          <Name>PartnerAttribute1</Name>
          <Value>PartnerAttribute1Value</Value>
        </PartnerAttribute>
        <PartnerAttribute>
          <Name>PartnerAttribute2</Name>
          <Value>PartnerAttribute2Value</Value>
        </PartnerAttribute>
      </partner>
      <account>
        <Name>AccountTest1</Name>
        <Number>123</Number>
        <AccountAttribute>
          <Name>AccountAttribute1</Name>
          <Value>AccountAttribute1Value</Value>
        </AccountAttribute>
        <AccountAttribute>
          <Name>AccountAttribute2</Name>
          <Value>AccountAttribute2Value</Value>
        </AccountAttribute>
        <PartnerName>PartnerTestName</PartnerName>
      </account>
    </tns:DXPartnerRequest>
  </soapenv:Body>
</ds:soapenv:Envelope>
DX_TPM_Partner Search Operation

The DX_TPM_Partner search operation searches the repository for a specified partner name.

Guidelines

Use the following guidelines when you search for a partner:

- The request must use the search operation.
- The search string must match the partner name exactly. For example, InformaticaCorp and Informa do not match the partner name Informatica. The partner name is not case sensitive.

Response

The web service returns a DXPartnerResponse object that includes the partner information in a DXPartner object.

DX_TPM_Partner Response

The DX_TPM_Partner web service returns NULL in the following situations:

- The partner name specified in the request does not exist in the repository.
- The partner name specified in the request is NULL or an empty string.
- The user account used to log in to the B2B Data Exchange Server is not authorized to view the partner.
- The name is already reserved for a dedicated system partner.

DX_TPM_Partner Update Operation

The DX_TPM_Partner update operation updates the information for a specified partner.

Guidelines

Use the following guidelines when you update a partner:

- The request must use the update operation.
- The search string must match the partner name exactly. For example, InformaticaCorp and Informa do not match the partner name Informatica. The partner name is not case sensitive.

Response

The web service returns a fault response in the following situations:

- The partner name specified in the request does not exist in the repository.
- The name is already reserved for a dedicated system partner.

Update SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Partner web service to update a partner:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 security" xmlns:tpm="http://b2b.informatica.com/dx/tpm/">
  <soapenv:Header>
    <sec:Security>
      <sec:UsernameToken>
        <Username>Administrator</Username>
        <Password>Administrator</Password>
      </UsernameToken>
    </sec:Security>
  </soapenv:Header>
  < TPM:Partner_configuration soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <TPM:Partner_configuration_id>sys</TPM:Partner_configuration_id>
  </TPM:Partner_configuration>
</soapenv:Envelope>
```
<sec:DXPassword DXPasswordType="PasswordText">sys</sec:DXPassword>

<!--<sec:Nonce>?<sec:Nonce>-->
<!--<sec:Created>?<sec:Created>-->
</sec:DXUsernameToken>
</sec:DXSecurity>
</soapenv:Header>
</soapenv:Body>
<tpm:DXPartnerRequest>
<operation>update</operation>
<partner>
<Name>PartnerTestName</Name>
<Description>PartnerTest Description</Description>
<!--Zero or more repetitions:--> 
<CategoryName>CategoryTest1</CategoryName>
<CategoryName>CategoryTest2</CategoryName>
<!--Zero or more repetitions:--> 
<ContactInformation>
>Name>PartnerTestContact1</Name>
<Description>PartnerTestContact1 Description</Description>
>Title>PartnerTestContact1 Title</Title>
<Address>PartnerTestContact1 Address</Address>
<PhoneNumber>+12345678912</PhoneNumber>
</BusinessPhoneNumber>
<FaxNumber>+12345678913</FaxNumber>
</EmailAddress>PartnerTestName1@informatica.com</EmailAddress>
</ContactInformation>

<ContactInformation>
>Name>PartnerTestContact2</Name>
<Description>PartnerTestContact2 Description</Description>
>Title>PartnerTestContact2 Title</Title>
<Address>PartnerTestContact2 Address</Address>
<PhoneNumber>+12345678922</PhoneNumber>
</BusinessPhoneNumber>
<FaxNumber>+12345678923</FaxNumber>
</EmailAddress>PartnerTestName2@informatica.com</EmailAddress>
</ContactInformation>

<!--Zero or more repetitions:--> 
<PartnerAttribute>
>Name>PartnerAttribute1</Name>
<Value>PartnerAttribute1Value</Value>
</PartnerAttribute>
<PartnerAttribute>
{Name>PartnerAttribute2</Name>
<Value>PartnerAttribute2Value</Value>
</PartnerAttribute>
</partner>
<account>
>Name>AccountTest1</Name>
<Number>123</Number>
<Description>AccountTest1 Description</Description>
<!--Zero or more repetitions:--> 
<CategoryName>CategoryTest3</CategoryName>
<CategoryName>CategoryTest4</CategoryName>
<!--Zero or more repetitions:--> 
<AccountAttribute>
{Name>AccountAttribute1</Name>
<Value>AccountAttribute1Value</Value>
</AccountAttribute>
<AccountAttribute>
{Name>PartnerName</Name>
</partner>
</account>
</tpm:DXPartnerRequest>
</soapenv:Body>
</soapenv:Envelope>
DX_TPM_Partner Delete Operation

The DX_TPM_Partner delete operation deletes a partner from the repository.

Guidelines

Use the following guidelines when you delete a partner:

- The request must use the delete operation.
- The search string must match the partner name exactly. For example, InformaticaCorp and Informa do not match the partner name Informatica. The partner name is not case sensitive.

Response

When the operation is successful, the web service sends an empty response to the client.

B2B Data Exchange ignores a delete request and does not return a fault response in the following situations:

- The partner name specified in the request does not exist in the repository.
- The partner name specified in the request is NULL or an empty string.
- The user account used to log in to the B2B Data Exchange Server is not authorized to delete the partner.

Delete SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Partner web service to delete a partner:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  security" xmlns:tpm="http://b2b.informatica.com/dx/tpm/">
  <soapenv:Header>
    <inf:Security>
      <UsernameToken>
        <Username>Administrator</Username>
        <Password>Administrator</Password>
      </UsernameToken>
    </inf:Security>
    <sec:DXSecurity>
      <sec:DXUsernameToken>
        <sec:DXUsername>sys</sec:DXUsername>
        <sec:DXPassword DXpasswordType="PasswordText">sys</sec:DXPassword>
      </sec:DXUsernameToken>
    </sec:DXSecurity>
  </soapenv:Header>
  <soapenv:Body>
    <tpm:DXPartnerRequest>
      <operation>delete</operation>
      <partnerName>PartnerTest</partnerName>
    </tpm:DXPartnerRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

DX_TPM_Partner Count Operation

The DX_TPM_Partner count operation returns the number of partners in the B2B Data Exchange repository.

Guidelines

Use the following guidelines when you count the partners in the repository:

- The request must use the count operation.

Response

The count operation returns the number of partners associated with the B2B Data Exchange user account in the message header.
Count SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Partner web service to count the number of partners in the Data Exchange repository:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  <soapenv:Header>
    <inf:Security>
      <UsernameToken>
        <Username>Administrator</Username>
        <Password>Administrator</Password>
      </UsernameToken>
    </inf:Security>
    <sec:DXSecurity>
      <sec:DXUsernameToken>
        <sec:DXUsername>sys</sec:DXUsername>
        <sec:DXPassword>DXPasswordType="PasswordText"</sec: DXPassword>
      </sec:DXUsernameToken>
    </sec:DXSecurity>
  </soapenv:Header>
  <tpm:DXPartnerRequest>
    <operation>count</operation>
  </tpm:DXPartnerRequest>
</soapenv:Body>
</soapenv:Envelope>
```

DX_TPM_Partner Get Operation

The DX_TPM_Partner get operation returns information for the partner associated with a specified index.

You can use the count and get operations together to get information on a partner by index. For example, you can get the total count of partners and use it to iterate through a "for" loop to get the information about each partner.

When you use the count as the basis for an index in an iteration, ensure that the count does not change until the iteration completes. If you insert or delete a partner during an iteration, the total number of partners can change and invalidate the index.

Guidelines

Use the following guidelines when you get information about a partner:

- The request must use the get operation.
- Partners are indexed by name in alphabetical order.
- The index numbers range from zero to the total count of partners minus one (0...n-1).

Response

The web service returns a DXPartnerResponse object that includes the partner information in a DXPartner object.

Get SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Partner web service to get information on a partner by index:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  <soapenv:Header>
    <inf:Security>
      <UsernameToken>
        <Username>Administrator</Username>
        <Password>Administrator</Password>
      </UsernameToken>
    </inf:Security>
  </soapenv:Header>
  <tpm:DXPartnerRequest>
    <operation>get</operation>
    <inf:DXPartnerRequest>/0</inf:DXPartnerRequest>
  </tpm:DXPartnerRequest>
</soapenv:Body>
</soapenv:Envelope>
```
DX_TPM_Account Web Service

The DX_TPM_Account web service creates, deletes, and manages accounts in the B2B Data Exchange repository.

Use the DX_TPM_Account web service to perform the following tasks:

- Create an account.
- Delete an account.
- Update the information for an account.
- Search for an account.
- Count the number of accounts in the repository.
- Get information for an account by index.

Reserved System Account

Do not use the name none when you create or update an account. This name is reserved for a dedicated system account. The web service returns a fault response if you try to assign this name to another account.

DX_TPM_Account Request

The DX_TPM_Account request defines the task for the DX_TPM_Account web service to perform.

The DX_TPM_Account request message includes the following elements:

- operation

Required. Type of operation the web service will perform. The web service can perform one of the following operations:

- create
- delete
- update
- search
- count
• get

account
Complex type object DXAccount that describes the partner account to create or update. Required by the create and update operations and ignored by other operations.

accountNumber
Account number to search or delete. Required by the search and delete operations and ignored by other operations.

partnerName
Name of the partner associated with the account that will be counted. Required by the count, get, search, and delete operations and ignored by other operations.

index
Index for the account to get. The index number starts at zero. Required by the get operation and ignored by other operations.

DX_TPM_Account Response
The DX_TPM_Account response is the reply from the DX_TPM_Account web service. The DX_TPM_Account response message includes the following elements:

account
Complex type object DXAccount that includes elements that describe the account to create or update. The account element is included in the response message for the following operations:

• create
• update
• get
• search

count
Number of accounts in the B2B Data Exchange repository. The count element is included in the response message for the count operation.

Note: When the web service request is a delete operation, the DX_TPM_Account web service sends an empty response to the client.

DX_TPM_Account Create Operation
The DX_TPM_Account create operation creates an account with a specified account name and number.

Guidelines
Use the following guidelines when you create an account:

• The request must use the create operation.

Response
The web service returns a DXAccountResponse object with the new account information in a DXAccount object.
The web service returns a fault response in the following situations:

- The DXAccount object has an element with an invalid value.
- The account already exists in the B2B Data Exchange repository.
- The account name or number identifies a dedicated system account.
- The partner for the account does not exist in the B2B Data Exchange repository.
- The associated partner name is reserved for a dedicated partner name or the account name is reserved for a dedicated system account.

Create SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Account web service to create an account:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:inf="http://www.informatica.com/"
    xmlns:sec="http://b2b.informatica.com/dx/
    security" xmlns:tpm="http://b2b.informatica.com/dx/tpm/">
    <soapenv:Header>
        <inf:Security>
            <inf:UsernameToken>
                <Username>Administrator</Username>
                <Password>Administrator</Password>
            </inf:UsernameToken>
        </inf:Security>
        <sec:DXSecurity>
            <sec:DXUsernameToken>
                <sec:DXUsername>sys</sec:DXUsername>
                <sec:DXPassword DXPasswordType="PasswordText">sys</sec:DXPassword>
            </sec:DXUsernameToken>
        </sec:DXSecurity>
    </soapenv:Header>
    <soapenv:Body>
        <tpm:DXAccountRequest>
            <operation>create</operation>
            <account>
                <Name>AccountTestName</Name>
                <Number>123</Number>
                <Description>AccountTest Description</Description>
                <CategoryName>CategoryTest1</CategoryName>
                <CategoryName>CategoryTest2</CategoryName>
                <AccountAttribute>
                    <Name>AccountAttribute1</Name>
                    <Value>AccountAttribute1Value</Value>
                </AccountAttribute>
                <AccountAttribute>
                    <Name>AccountAttribute2</Name>
                    <Value>AccountAttribute2Value</Value>
                </AccountAttribute>
                <PartnerName>PartnerTestName</PartnerName>
            </account>
        </tpm:DXAccountRequest>
    </soapenv:Body>
</soapenv:Envelope>
```
DX_TPM_Account Search Operation

The DX_TPM_Account search operation searches the repository for a specified partner name and account number.

Guidelines

Use the following guidelines when you search for an account:

- The request must use the search operation.
- The partner name and account number must exactly match the partner name and account number of the account to search for.

Response

The web service returns a DXAccountResponse object that includes the account information in a DXAccount object.

The web service returns NULL in the following situations:

- An account with the partner name and account number specified in the request does not exist in the repository.
- The partner name or account number specified in the request is NULL or an empty string.
- The user account used to log in to the B2B Data Exchange Server is not authorized to view the account.
- The partner name or account number identifies a reserved system partner or account.

Search SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Account web service to search for an account:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:inf="http://www.informatica.com/
xmlns:sec="http://b2b.informatica.com/dx/security"
xmlns:tpm="http://b2b.informatica.com/dx/tpm">
  <soapenv:Header>
    <inf:Security>
      <inf:UsernameToken>
        <inf:Username>Administrator</inf:Username>
        <inf:Password>Administrator</inf:Password>
      </inf:UsernameToken>
    </inf:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tpm:DXAccountRequest>
      <operation>search</operation>
      <partnerName>myPartner</partnerName>
      <accountNumber>myAccount</accountNumber>
    </tpm:DXAccountRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
DX_TPM_Account Update Operation

The DX_TPM_Account update operation updates the information for a specified account number.

Guidelines

Use the following guidelines when you update an account:

- The request must use the update operation.

Response

The web service returns a fault response in the following situations:

- The account number specified in the request does not exist in the repository.
- The account number identifies a reserved system account.

Update SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Account web service to update an account:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:inf="http://www.informatica.com/"
xmlns:sec="http://b2b.informatica.com/dx/security"
xmlns:tpm="http://b2b.informatica.com/dx/tpm/>
<soapenv:Header>
  <inf:Security>
    <inf:UsernameToken>
      <inf:Username>Administrator</inf:Username>
      <inf:Password>Administrator</inf:Password>
    </inf:UsernameToken>
  </inf:Security>
  <sec:DXSecurity>
    <sec:DXUsernameToken>
      <sec:DXUsername>sys</sec:DXUsername>
      <sec:DXPasswordType="PasswordText">sys</sec:DXPassword>
    </sec:DXUsernameToken>
  </sec:DXSecurity>
</soapenv:Header>
<soapenv:Body>
<tpm:DXAccountRequest>
  <operation>update</operation>
  <account>
    <Name>AccountTestName</Name>
    <Number>123</Number>
    <Description>AccountTest Description</Description>
    <!-- Zero or more repetitions -->
    <CategoryName>CategoryTest1</CategoryName>
    <CategoryName>CategoryTest2</CategoryName>
    <!-- Zero or more repetitions -->
    <AccountAttribute>
      <Name>AccountAttribute1</Name>
      <Value>AccountAttribute1Value</Value>
    </AccountAttribute>
    <AccountAttribute>
      <Name>AccountAttribute2</Name>
      <Value>AccountAttribute2Value</Value>
    </AccountAttribute>
    <PartnerName>PartnerTestName</PartnerName>
  </account>
</tpm:DXAccountRequest>
</soapenv:Body>
</soapenv:Envelope>
```
DX_TPM_Account Delete Operation

The DX_TPM_Account delete operation deletes an account from the repository.

Guidelines

Use the following guidelines when you delete an account:

- The request must use the delete operation.
- Specify a partner name and account number that exactly matches the partner name and account number of the account to delete.

Response

When the operation is successful, the web service sends an empty response to the client.

B2B Data Exchange ignores a delete request and does not return a fault response in the following situations:

- The partner name and account number of the account specified in the request does not exist in the repository.
- The partner name or account number specified in the request is NULL or an empty string.
- The user account used to log in to the B2B Data Exchange Server is not authorized to delete the account.

Delete SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Account web service to delete an account:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
security" xmlns:tpm="http://b2b.informatica.com/dx/tpm/"
<soapenv:Header>
   <inf:Security>
      <UserNameToken>
         <Username>Administrator</Username>
         <Password>Administrator</Password>
      </UserNameToken>
   </inf:Security>
   <sec:DXSecurity>
      <sec:DXUsernameToken>
         <sec:DXUsername>syst</sec:DXUsername>
         <sec:DXPassword DXPasswordType="PasswordText">sys</sec:DXPassword>
      </sec:DXUsernameToken>
   </sec:DXSecurity>
</soapenv:Header>
<soapenv:Body>
   <tpm:DXAccountRequest>
      <operation>delete</operation>
      <partnerName>myPartner</partnerName>
      <accountNumber>myAccount</accountNumber>
   </tpm:DXAccountRequest>
</soapenv:Body>
</soapenv:Envelope>
```

DX_TPM_Account Count Operation

The DX_TPM_Account count operation returns the number of accounts associated with a specified partner.

Guidelines

Use the following guidelines when you count the accounts for a partner:

- The request must use the count operation.
Response

The web service returns a fault response in the following situations:

- The partner name does not exist in the B2B Data Exchange repository.
- The partner name is NULL or an empty string, or exceeds the maximum length.
- The partner name identifies a reserved system partner.

Count SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Account web service to count the accounts for a partner:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <soapenv:Header>
    <sec:DXSecurity>
      <sec:DXUsernameToken>
        <sec:DXUsername>sys</sec:DXUsername>
      </sec:DXUsername>
      <sec:DXPassword DXPasswordType="PasswordText">sys</sec:DXPassword>
    </sec:DXSecurity>
  </soapenv:Header>
  <tpm:DXAccountRequest>
    <operation>count</operation>
    <partnerName>myPartner</partnerName>
  </tpm:DXAccountRequest>
</soapenv:Envelope>
```

DX_TPM_Account Get Operation

The DX_TPM_Account get operation returns information for the account associated with a specified index.

You can use the count and get operations together to get information on the account by index. For example, you can get the total number of accounts for a partner and use it to iterate through a “for” loop to get the information about each account.

When you use the count as the basis for an index in an iteration, ensure that the count does not change until the iteration completes. If you insert or delete an account during an iteration, the total number of accounts can change and invalidate the index.

Guidelines

Use the following guidelines when you get information about an account:

- The request must use the get operation.
- Accounts are indexed by account name in alphabetical order.
- The index numbers range from zero to the total number of accounts minus one (0 ... n-1).

Response

The web service returns a DXAccountResponse object that includes the account information in a DXAccount object.
The web service returns a fault response in the following situations:

- The index is outside the range of index numbers.
- The partner name does not exist in the B2B Data Exchange repository.
- The partner name identifies a dedicated system partner or account.

Get SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Account web service to get an account by index:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    security" xmlns:tpm="http://b2b.informatica.com/dx/tpm/"
>
  <soapenv:Header>
    <inf:Security>
      <inf:UsernameToken>
        <Username>Administrator</Username>
        <Password>Administrator</Password>
      </inf:UsernameToken>
    </inf:Security>
    <sec:DXSecurity>
      <sec:DXUsernameToken>
        <sec:DXUsername>sys</sec:DXUsername>
        <sec:DXPassword>DXPasswordType="PasswordText">sys</sec:DXPassword>
      </sec:DXUsernameToken>
    </sec:DXSecurity>
  </soapenv:Header>
  <soapenv:Body>
    <tpm:DXAccountRequest>
      <operation>get</operation>
      <partnerName>myPartner</partnerName>
      <index>0</index>
    </tpm:DXAccountRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

**DX_TPM_Profile Web Service**

The DX_TPM_Profile web service creates, deletes, and manages profiles in the B2B Data Exchange repository.

Use the DX_TPM_Profile web service to perform the following tasks:

- Create a profile.
- Delete a profile.
- Update the information for a profile.
- Search for a profile.
- Count the profiles in the repository.
- Get the information for a profile by index.

**Reserved System Profile**

Do not use the name **none** when you create or update a profile. This name is reserved for a dedicated system profile. The web service returns a fault response if you try to assign this name to another profile.
DX_TPM_Profile Request

The DX_TPM_Profile request defines the task performed by the DX_TPM_Profile web service.

The DX_TPM_Profile request message includes the following elements:

operation

Required. Type of operation the web service will perform. The web service can perform one of the following operations:

• create
• delete
• update
• search
• count
• get

profile

Complex type object DXProfile that describes the profile to create or update. Required by the create and update operations and ignored by other operations.

profileName

Name of the profile to search or delete. Required by the search and delete operations.

index

Index of the profile to get. The index number starts at zero. Required by the get operation.

DX_TPM_Profile Response

The DX_TPM_Profile response message is the reply from the DX_TPM_Profile web service.

The DX_TPM_Profile response message includes the following elements:

profile

Complex type object DXProfile that describes the trading partner to create or update. The profile element is included in the response message for the following operations:

• create
• update
• get
• search

count

Number of profiles in the B2B Data Exchange repository. The count element is included in the response message for the count operation.

Note: When the web service request is a delete operation, the DX_TPM_Profile web service sends an empty response to the client.
**DX_TPM_Profile Create Operation**

The DX_TPM_Profile `create` operation creates a profile with a specified profile name.

**Guidelines**

Use the following guidelines when you create a profile:

- The request must use the `create` operation.

**Response**

The web service returns a DXProfileResponse object with the new profile information in a DXProfile object.

The web service returns a fault response in the following situations:

- The DXProfile object has an element or attribute with an invalid value.
- The profile name already exists in the B2B Data Exchange repository.
- The partner name is not null but the account number is null.
- The account number is not null but the partner name is null.
- The profile name identifies a reserved system profile.

**Create SOAP Message Example**

The following SOAP message shows an example of a request to the DX_TPM_Profile web service to create a profile:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:inf="http://www.informatica.com/"
xmlns:sec="http://b2b.informatica.com/dx/security"
xmlns:tpm="http://b2b.informatica.com/dx/tpm/">
 <soapenv:Header>
   <inf:Security>
     <UsernameToken>
       <UserName>Administrator</UserName>
       <Password>Administrator</Password>
     </UsernameToken>
   </inf:Security>
   <sec:DXSecurity>
     <sec:DXUsernameToken>
       <sec:DXUsername>sys</sec:DXUsername>
       <sec:DXPassword DXPasswordType="PasswordText"/>sys</sec:DXPassword>
     </sec:DXUsernameToken>
   </sec:DXSecurity>
 </soapenv:Header>
 <soapenv:Body>
   <tpm:DXProfileRequest>
     <operation>create</operation>
     <profile>
       <id>123</id>
       <Name>ProfileTestName</Name>
       <Description>ProfileTestName Description</Description>
       <PartnerName>PartnerTestName</PartnerName>
       <AccountNumber>123</AccountNumber>
       <WorkflowName>WORKFLOW_TEST_NAME</WorkflowName>
       <ScheduleName>ScheduleTestName</ScheduleName>
       <Status>ENABLED</Status>
   </profile>
   </tpm:DXProfileRequest>
 </soapenv:Body>
</soapenv:Envelope>
```
The DX_TPM_Profile search operation searches the repository for a specified profile.

Guidelines

Use the following guidelines when you search for a profile:

- The request must use the search operation.
- The search string must match the profile name exactly. For example, EDIWithAckn and Acknowledgement do not match the profile name EDIWithAcknowledgement. The profile name is not case sensitive.

Response

The web service returns a DXProfileResponse object with the new profile information in a DXProfile object.

The web service returns NULL in the following situations:

- The profile name specified in the request does not exist in the repository.
- The profile name specified in the request is NULL or an empty string.
- The user account used to log in to the B2B Data Exchange Server is not authorized to view information about the profile.
- The profile name identifies a reserved system profile.

Search SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Profile web service to search for a profile:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Header>
    <inf:Security>
      <sec:DXUserNameToken>
        <sec:DXUserName>sys</sec:DXUserName>
        <sec:DXPassword>DXPasswordType="PasswordText"</sec:DXPassword>
      </sec:DXUserNameToken>
    </inf:Security>
  </soapenv:Header>
  <tpm:DXProfileRequest>
    <operation>search</operation>
    <profileName>myProfile</profileName>
  </tpm:DXProfileRequest>
</soapenv:Envelope>
```
DX_TPM_Profile Update Operation

The DX_TPM_Profile update operation updates the information for a specified profile.

Guidelines

Use the following guidelines when you update a profile:

- The request must use the update operation.

Response

The web service returns a fault response in the following situations:

- The profile name specified in the request does not exist in the repository.
- The profile name identifies a reserved system profile.

Update SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Profile web service to update a profile:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
<soapenv:Header>
  <inf:Security>
    <UserNameToken>
      <UserName>Administrator</UserName>
      <Password>Administrator</Password>
    </UserNameToken>
  </inf:Security>
  <sec:DXSecurity>
    <sec:DXUsernameToken>
      <sys:DXUsername>sys</sys:DXUsername>
      <sec:DXPasswordType="PasswordText">sys</sec:DXPassword>
    </sec:DXUsernameToken>
  </sec:DXSecurity>
</soapenv:Header>
<soapenv:Body>
  <tpm:DXProfileRequest>
    <operation>update</operation>
    <profile>
      <Id>123</Id>
      <Name>ProfileTestName</Name>
      <Description>ProfileTestName_Description</Description>
      <PartnerName>PartnerTestName</PartnerName>
      <AccountNumber>123</AccountNumber>
      <WorkflowName>WORKFLOW_TEST_NAME</WorkflowName>
      <ScheduleName>ScheduleTestName</ScheduleName>
      <Status>ENABLED</Status>
    <!-- Zero or more repetitions:-->
    <WorkflowParameter>
      <Name>WorkflowTempParameter1</Name>
      <Value>WorkflowTempParameter1Value</Value>
    </WorkflowParameter>
    <WorkflowParameter>
      <Name>WorkflowTempParameter2</Name>
      <Value>WorkflowTempParameter2Value</Value>
    </WorkflowParameter>
    <!-- Zero or more repetitions:-->
    <CategoryName>CategoryName1</CategoryName>
    <CategoryName>CategoryName2</CategoryName>
    <!-- Zero or more repetitions:-->
  </profile>
</tpm:DXProfileRequest>
</soapenv:Body>
</soapenv:Envelope>
```
DX_TPM_Profile Delete Operation

The DX_TPM_Profile delete operation deletes a profile from the repository.

Guidelines

Use the following guidelines when you delete a profile:

- The request must use the delete operation.

Response

When the request is successful, the web service sends an empty response to the client.

B2B Data Exchange ignores a delete request and does not return a fault response in the following situations:

- The profile does not exist in the repository.
- The profile name specified in the request is NULL or an empty string.
- The user account used to log in to the B2B Data Exchange Server is not authorized to delete the profile.

Delete SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Profile web service to delete a profile:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
security" xmlns:tpm="http://b2b.informatica.com/dx/tpm/">
  <soapenv:Header>
    <inf:Security>
      <soapenv:UsernameToken>
        <Username>Administrator</Username>
        <Password>Administrator</Password>
      </soapenv:UsernameToken>
      <sec:DXSecurity>
        <sec:DXUsernameToken>
          <sec:DXUsername>sys</sec:DXUsername>
          <sec:DXPassword DXPasswordType="PasswordText">sys</sec:DXPassword>
        </sec:DXUsernameToken>
      </sec:DXSecurity>
    </inf:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tpm:DXProfileRequest>
      <operation>delete</operation>
      <profileName>myProfile</profileName>
    </tpm:DXProfileRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
DX_TPM_Profile Count Operation

The DX_TPM_Profile count operation returns the number of profiles in the B2B Data Exchange repository.

Guidelines

Use the following guidelines when you count the profiles in the repository:

- The request must use the count operation.

Response

The count operation returns the number of profiles associated with the B2B Data Exchange user account in the message header.

Count SOAP Message Example

The following SOAP message shows an example of a request to the DX_TPM_Profile web service to count the profiles in the repository:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:inf="http://www.informatica.com/
xmlns:sec="http://b2b.informatica.com/dx/
xmlns:tpm="http://b2b.informatica.com/dx/tpm/">
  <soapenv:Header>
    <inf:Security>
      <sec:DXUsernameToken>
        <sec:DXUsername>sys</sec:DXUsername>
        <sec:DXPassword DXPasswordType="PasswordText">sys</sec:DXPassword>
      </sec:DXUsernameToken>
    </inf:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tpm:DXProfileRequest>
      <operation>count</operation>
    </tpm:DXProfileRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

DX_TPM_Profile Get Operation

The DX_TPM_Profile get operation returns information for the profile associated with a specified index.

You can use the count and get operations together to get information on a profile by index. For example, you can get the total count of profiles and use it to iterate through a "for" loop to get the information about each profile.

When you use the count as the basis for an index in an iteration, ensure that the count does not change until the iteration completes. If you insert or delete a profile during an iteration, the total number of profiles can change and invalidate the index.

Guidelines

Use the following guidelines when you get information about a profile:

- The request must use the get operation.
- Profiles are indexed by profile name in alphabetical order.
- The index numbers range from zero to the total count of profiles minus one (0...n-1).
Response
The web service returns a DXProfileResponse object that includes the profile information in a DXProfile object.
The web service sends a fault message to the client if the index is outside the range of index numbers.

Get SOAP Message Example
The following SOAP message shows an example of a request to the DX_TPM_Profile web service to get a profile by index:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tpm="http://b2b.informatica.com/dx/tpm/">
  <soapenv:Header>
    <inf:Security>
      <inf:UsernameToken>
        <inf:Username>Administrator</inf:Username>
        <inf:Password>Administrator</inf:Password>
      </inf:UsernameToken>
    </inf:Security>
    <sec:DXSecurity>
      <sec:DXUsernameToken>
        <sec:DXUsername>sys</sec:DXUsername>
        <sec:DXPassword DXPasswordType="PasswordText">sys</sec:DXPassword>
      </sec:DXUsernameToken>
    </sec:DXSecurity>
  </soapenv:Header>
  <soapenv:Body>
    <tpm:DXProfileRequest>
      <operation>get</operation>
      <index>0</index>
    </tpm:DXProfileRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

DX_Endpoint Web Service
The DX_Endpoint web service sends a document to B2B Data Exchange and routes it to a profile for processing.

DX_Endpoint Request
The DX_Endpoint request message defines a document for the DX_Endpoint web service to process.
The DX_Endpoint request message includes the following elements:

Data
Required. Document to be processed. The document must have Base64 encoding.

DXProperty
Key-value pair that contains the properties of the document. This element is defined by the complex type object DXProperty and can occur multiple times.

You can include the following keys:
- application. Name of the application to use for routing the document to a profile for processing.
- profileName. Name of the profile to use for routing the document to a profile for processing.
- partnerName. Name of the partner to use for routing the document to a profile for processing.
- accountNumber. Account number to use for routing the document to a profile for processing.
- accountName. Name of the account to use for routing the document to a profile for processing.
- dataByReference. Indicates whether the request contains data or a reference to data. Set to true when the request contains a reference to the data. Set to false when the request contains the data.
- mimetype. The MIME type of the data, such as text/plain or text/html.

**CustomProperty**

Key-value pair that contains custom properties of the document. This element can occur multiple times.

### DX_Endpoint Response

The DX_Endpoint response defines the reply from the DX_Endpoint web service.

The DX_Endpoint web service sends an empty response to the client when the request completes successfully.

### Complex Type Objects in the Data Exchange Web Services

The WSDL for the Data Exchange web services include the following complex type objects:

- DXPartner. Defines a partner.
- DXAccount. Defines an account.
- DXProfile. Defines a profile.
- DXContactInformation. Defines the contact information for a partner.

The properties of the elements in the complex type objects must match the properties of the corresponding data in B2B Data Exchange. For example, the partner name element has the same maximum length as a partner in the B2B Data Exchange repository.

### DXPartner Object

The DXPartner object defines a partner.

The following table describes the elements and attributes in the DXPartner object:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Unique identifier for the partner. Generated by B2B Data Exchange when a partner is created.</td>
</tr>
<tr>
<td>Name</td>
<td>Required. Name of the partner.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the partner.</td>
</tr>
</tbody>
</table>
### DXAccount Object

The DXAccount object defines an account.

The following table describes the elements and attributes in the DXAccount object:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>Required. Account number.</td>
</tr>
<tr>
<td>Name</td>
<td>Required. Name of the account.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the account.</td>
</tr>
<tr>
<td>CategoryName</td>
<td>Category of the account. This element can occur multiple times.</td>
</tr>
<tr>
<td>AccountAttribute</td>
<td>DXAccountAttribute object that consists of a name and attribute value pair.</td>
</tr>
<tr>
<td>PartnerName</td>
<td>Required. Name of the partner associated with the account.</td>
</tr>
<tr>
<td>CreationDate</td>
<td>Date when the account is created. Generated by the B2B Data Exchange Server</td>
</tr>
<tr>
<td>LastUpdatedDate</td>
<td>Date when the account was last updated. Generated by the B2B Data Exchange</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CategoryName</td>
<td>Category of the partner. This element can occur multiple times.</td>
</tr>
<tr>
<td>ContactInformation</td>
<td>DXContactInformation object that contains the details of the partner contact.</td>
</tr>
<tr>
<td></td>
<td>This element can occur multiple times.</td>
</tr>
<tr>
<td>PartnerAttribute</td>
<td>DXPartnerAttribute object that consists of a name and attribute value pair.</td>
</tr>
<tr>
<td></td>
<td>This element can occur multiple times.</td>
</tr>
<tr>
<td>CreationDate</td>
<td>Date when the partner is created. Generated by the B2B Data Exchange Server</td>
</tr>
<tr>
<td></td>
<td>when it creates the partner.</td>
</tr>
<tr>
<td>LastUpdatedDate</td>
<td>Date when the partner was last updated. Generated by the B2B Data Exchange</td>
</tr>
<tr>
<td></td>
<td>Server when it updates the partner.</td>
</tr>
</tbody>
</table>
DXProfile Object

The DXProfile object defines a profile.

The following table describes the elements and attributes in the DXProfile object:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Unique identifier for the profile. Generated by B2B Data Exchange when a profile is created.</td>
</tr>
<tr>
<td>Name</td>
<td>Required. Name of the profile.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the profile.</td>
</tr>
<tr>
<td>PartnerName</td>
<td>Name of the partner associated with the account.</td>
</tr>
<tr>
<td>AccountNumber</td>
<td>Account number.</td>
</tr>
<tr>
<td>WorkflowName</td>
<td>Required. Name of the workflow associated with the profile.</td>
</tr>
<tr>
<td>ScheduleName</td>
<td>Name of the schedule associated with the profile.</td>
</tr>
<tr>
<td>Status</td>
<td>Required. DXProfileStatus object that describes the status of the profile.</td>
</tr>
<tr>
<td></td>
<td>Set this element to one of the following statuses:</td>
</tr>
<tr>
<td></td>
<td>- ENABLED</td>
</tr>
<tr>
<td></td>
<td>- DISABLED</td>
</tr>
<tr>
<td>WorkflowParameter</td>
<td>Parameters for the workflow associated with the profile. The DXWorkflowParameter object consists of a name and attribute value pair. This element can occur multiple times.</td>
</tr>
<tr>
<td>CategoryName</td>
<td>Category of the profile. This element can occur multiple times.</td>
</tr>
<tr>
<td>EventAttribute</td>
<td>Event attribute associated with the profile. The DXEventAttribute object consists of a name and attribute value pair. This element can occur multiple times.</td>
</tr>
<tr>
<td>CreationDate</td>
<td>Date when the partner is created. Generated by the B2B Data Exchange Server when it creates the profile.</td>
</tr>
<tr>
<td>LastUpdatedDate</td>
<td>Date when the partner was last updated. Generated by the B2B Data Exchange Server when it updates the profile.</td>
</tr>
</tbody>
</table>

DXContactInformation Object

The DXContactInformation object defines the contact information for a partner.

The following table describes the elements and attributes in the DXContactInformation object:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Required. Name of the contact.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the contact.</td>
</tr>
<tr>
<td>Title</td>
<td>Title of the contact.</td>
</tr>
</tbody>
</table>
### General B2B Data Exchange Web Service Behavior

The B2B Data Exchange web services cannot modify a dedicated system profile, account, or partner, or properties that identify these objects.

The B2B Data Exchange web services have the following general characteristics and behavior:

- The web service operations do not retrieve or modify a dedicated system profile, account, or partner.
- The web service operations cannot modify a property that is used as an identifier. For example, you cannot use the B2B Data Exchange web services to modify a partner name or an account number.
- The web services return fault responses in the following situations:
  - An internal error occurs when a web service is running.
  - A required parameter in the web service request is empty or set to NULL.
  - An element in the DXPartner, DXAccount, or DXProfile object has a value that is not valid or exceeds the maximum length.

### B2B Data Exchange Web Service Security

When you send a request to a B2B Data Exchange web service, you must provide the user credentials to access the PowerCenter repository and the B2B Data Exchange server. The user account for the B2B Data Exchange server determines the web service access to the objects in the B2B Data Exchange repository.

#### User Authentication

A client application that calls a B2B Data Exchange web service must log in to the PowerCenter repository and the B2B Data Exchange server to perform any operation.

You must provide the user credentials for the PowerCenter repository and for the B2B Data Exchange server in the SOAP message header of the web service request.
The message header contains the following security elements:

- **Security.** Contains the user name and password to log in to the PowerCenter repository that stores the B2B Data Exchange web services. The user account must have permissions to run the B2B Data Exchange web services. You canplain text passwords and digested passwords. For more information about configuring digested passwords in PowerCenter, see the Informatica PowerCenter Web Services Provider Guide.

- **DXSecurity.** Contains the user name and password of the user account to log in to the B2B Data Exchange server. The user account must have permissions to view objects in the B2B Data Exchange repository. The DXSecurity element can have one of the following passwords:
  - Plain text password. Password in plain text.
  - Digested password. Encrypted password that is hashed with a nonce value and a time stamp.

The password element has a password type attribute that indicates the type of password security used to log in to the B2B Data Exchange server. Set the password type attribute to PasswordText for a plain text password or to PasswordDigest for a digested password. If the password type attribute is omitted, the password type defaults to PasswordText.

**Note:** Although a B2B Data Exchange user name can include a slash (/), a user account with a slash in the name is not valid for B2B Data Exchange web services.

The following example shows the message header for a B2B Data Exchange web service request with the DXSecurity and Security elements:

```xml
<soap:Header>
  <ns0:DXSecurity xmlns:ns0="http://b2b.informatica.com/dx/security">
    <ns0:DXUsernameToken>
      <ns0:DXUsername>[string]</ns0:DXUsername>
      <ns0:DXPassword DXPasswordType="[undefined]">[string]</ns0:DXPassword>
      <ns0:Nonce>[string]</ns0:Nonce>
      <ns0:Created>[string]</ns0:Created>
    </ns0:DXUsernameToken>
  </ns0:DXSecurity>
  <ns0:Security xmlns:ns0="http://www.informatica.com/">
    <UsernameToken>
      <Username>[string]</Username>
      <Password>[string]</Password>
    </UsernameToken>
  </ns0:Security>
</soap:Header>
```

**User Access**

Permissions define the level of access to B2B Data Exchange objects that a user account can have. The permissions for the user account in the DXSecurity element of the web service request header determines access to the objects in the B2B Data Exchange repository.

For example, the B2B Data Exchange repository has a hundred partners and user AJones has permission on five of the partners. A DX_TPM_Partner web service request that uses the credentials of AJones in the DXSecurity element and performs a count operation returns five partners instead of a hundred.
This chapter includes the following topics:

- B2B Data Exchange Run-time Java API Overview, 93
- API Reference, 94

B2B Data Exchange Run-time Java API Overview

Use the B2B Data Exchange Run-time API to access the B2B Data Exchange processes and repository. When you develop workflows in PowerCenter, you can use the B2B Data Exchange transformations to process B2B Data Exchange documents. The transformations call the B2B Data Exchange Run-time API methods to process the documents. You can create additional transformations and use the methods in the B2B Data Exchange Run-time API to define the transformation behavior. You can also call the methods from other processes or applications based on your document processing requirements.

The B2B Data Exchange Run-time API contains the following package:

    com.informatica.b2b.dx.client.DXClient

All B2B Data Exchange Run-time API methods that create a database transaction commit the transaction.

For more information about the B2B Data Exchange Run-time Java API, see the online API reference located in: <DXInstallationDir>/powercenter/javadoc/dxserver/apidocs/index.html

B2B Data Exchange Run-time Java API Versions

The B2B Data Exchange Run-time Java API includes a client class for each B2B Data Exchange version. To use the B2B Data Exchange Run-time API, include the following line of code in your client application code:

    DXClient dxClient = DXClientFactory.getClient(DXClientTypes.POWERCENTER_CLIENT);

Then, use the DXClient object when you call an API method. For example:

    dxClient.flowEntry(profileId, reuseEventId);

By default, the DXClientFactory returns the latest version of the B2B Data Exchange API.

To use the version 8.6.1 of the API, perform one of the following tasks:

- Modify your client code. Use the following lines of code in your client application code:

  ```java
  DXClient861 dxClient = (DXClient861)
  DXClientFactory.getClient(DXClientTypes.POWERCENTER_CLIENT);
  ```
• **Add the API 8.6.1 jar file to the Java SDK Classpath of the Integration Service.** In the Administrator Tool, edit the Integration Service that runs the Data Exchange workflows. Add the following jar file before all the other jar files in the Java SDK Classpath:

```
<DataExchangeInstallationDir>/powercenter/lib/dx-face-api-8.6.1-
<DataExchangeVersion>.jar;
```

For more information about the methods in the previous version of the B2B Data Exchange Run-time Java API, see the online API reference:

```
<DXInstallationDIR>\powercenter\javadoc\apidocs\com\informatica\b2b\dx\client
\DXClient861.html
```

The online reference for B2B Data Exchange API 8.6.1 contains information about how to upgrade your code to use a later B2B Data Exchange Run-time Java API version.

# API Reference

The following table lists the methods available in the B2B Data Exchange Run-time API:

<table>
<thead>
<tr>
<th>Method</th>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addEventDocument ( String eventId, Document document )</code></td>
<td>String</td>
<td>Attaches a document to an event and returns the document ID for the document attached to the event.</td>
</tr>
<tr>
<td><code>completeCorrelation ( String eventId, String correlationId )</code></td>
<td>String</td>
<td>Completes a reconciliation process and returns the ID of the event that initiated the reconciliation process.</td>
</tr>
<tr>
<td><code>completeCorrelations ( String eventId )</code></td>
<td>void</td>
<td>Closes all correlation events of the specified event Id.</td>
</tr>
<tr>
<td><code>createChildEvent ( String parentEventId, String profileId )</code></td>
<td>String</td>
<td>Creates a child event based on a profile ID and returns the event ID of the child event.</td>
</tr>
<tr>
<td><code>createChildEvent ( String parentEventId, String profileId, String eventStatusName, String eventTypeName, String subject )</code></td>
<td>String</td>
<td>Creates a child event based on a profile ID and sets the properties of the child event. Returns the event ID of the child event.</td>
</tr>
<tr>
<td><code>flowEntry ( String profileId )</code></td>
<td>String</td>
<td>Indicates the start of a workflow. This method creates an event and returns the event ID.</td>
</tr>
<tr>
<td>Method</td>
<td>Return Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>flowEntry</code> ( String profileId, String reuseeventId )</td>
<td>String</td>
<td>Indicates the start of a workflow. You can specify the event ID to use.</td>
</tr>
<tr>
<td><code>flowExit</code> ( String eventId )</td>
<td>String</td>
<td>Indicates the end of a workflow. This method updates the status of the event with the system default status and returns the event ID.</td>
</tr>
<tr>
<td><code>flowExit</code> ( String eventId, String status )</td>
<td>String</td>
<td>Indicates the end of a workflow. This method updates the status of the event with the given status and returns the event ID.</td>
</tr>
<tr>
<td><code>generateTemporaryFilePath</code> ( String eventId )</td>
<td>String</td>
<td>Creates a unique file name in the B2B Data Exchange temporary document directory and returns the path and file name.</td>
</tr>
<tr>
<td><code>getChildrenCount</code> ( String parentEventId )</td>
<td>Integer</td>
<td>Returns the number of child events for a parent event.</td>
</tr>
<tr>
<td><code>getDocumentData</code> ( Document document, Long offset, Long count, String eventId )</td>
<td>byte[]</td>
<td>Returns the data of a document.</td>
</tr>
<tr>
<td><code>getDocumentFilePath</code> ( Document documentReference, String eventId )</td>
<td>String</td>
<td>Returns the path and file name of the file associated with a document reference.</td>
</tr>
<tr>
<td><code>getEventAttribute</code> ( String eventId, String attributeKey )</td>
<td>String</td>
<td>Returns the value of an event attribute.</td>
</tr>
<tr>
<td><code>getEventContext</code> ( String eventId )</td>
<td>Map</td>
<td>Returns the run-time context of an event.</td>
</tr>
<tr>
<td><code>getEventDocuments</code> ( String eventId )</td>
<td>List&lt;Document&gt;</td>
<td>Returns information about all documents associated with an event.</td>
</tr>
<tr>
<td><code>getEventInfo</code> ( String eventId )</td>
<td>Properties</td>
<td>Returns the information associated with an event, including event status and event type.</td>
</tr>
<tr>
<td><code>getEventParentEventId</code> ( String eventId )</td>
<td>String</td>
<td>Returns the event ID of the parent event.</td>
</tr>
<tr>
<td><code>getEventStatusName</code> ( String eventId )</td>
<td>String</td>
<td>Returns the status of an event.</td>
</tr>
<tr>
<td>Method</td>
<td>Return Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>incrementProfileSequenceValue (</td>
<td>String</td>
<td>Increments by one the value of a sequence parameter associated with a profile.</td>
</tr>
<tr>
<td>String profileId,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String parameterSequenceId</td>
<td></td>
<td></td>
</tr>
<tr>
<td>incrementProfileSequenceValue (</td>
<td>String</td>
<td>Increments by incrementBy the value of a sequence parameter associated with a profile.</td>
</tr>
<tr>
<td>String profileId,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String parameterSequenceId,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String incrementBy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>initiateCorrelation (</td>
<td>void</td>
<td>Initiates a reconciliation process.</td>
</tr>
<tr>
<td>String eventId,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String correlationId,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String correlationType,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>long timeWindowSeconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>releaseDelayedEvents (</td>
<td>void</td>
<td>Releases delayed events.</td>
</tr>
<tr>
<td>String profileId,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>List&lt;String&gt; eventIds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>resolveProfileId (</td>
<td>String</td>
<td>Returns the profile ID associated with an application and an account number or partner.</td>
</tr>
<tr>
<td>String accountNumber,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String application,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String partnerName,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String profileName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>setEventAttribute (</td>
<td>String</td>
<td>Sets the value of the attribute associated with an event.</td>
</tr>
<tr>
<td>String eventId,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String attributeKey,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String attributeValue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>setEventStatus (</td>
<td>void</td>
<td>Updates the status of an event.</td>
</tr>
<tr>
<td>String eventId,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String eventStatus</td>
<td></td>
<td></td>
</tr>
<tr>
<td>setEventStatus (</td>
<td>void</td>
<td>Updates the status of an event.</td>
</tr>
<tr>
<td>String eventId,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String eventStatusId,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String comments</td>
<td></td>
<td></td>
</tr>
<tr>
<td>setEventSubject (</td>
<td>void</td>
<td>Updates the subject of an event.</td>
</tr>
<tr>
<td>String eventId,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>String subject</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Return Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>setEventType (String eventId, String eventType)</td>
<td>void</td>
<td>Updates the type of an event.</td>
</tr>
<tr>
<td>setProfileSequenceValue (String profileId, String parameterSequenceId, Object newValue)</td>
<td>String</td>
<td>Assigns a value to a sequence parameter associated with a profile.</td>
</tr>
</tbody>
</table>
B2B Data Exchange Forms Designer Java API

This chapter includes the following topics:

- B2B Data Exchange Forms Designer Java API Overview, 98
- B2B Data Exchange Forms Designer Java API Reference, 99

B2B Data Exchange Forms Designer Java API Overview

Use the B2B Data Exchange Forms Designer Java API to manage values for partner attributes, account attributes, and workflow parameters. When you use the Forms Designer, you can populate lists of possible values from external sources. You can also set default values for the lists. B2B Data Exchange throws an exception if any error occurs while the lists are populated.

The B2B Data Exchange Forms Designer Java API contains the following packages:

- com.informatica.b2b.dx.api.exception. Contains exceptions to throw if any error occurs while the Forms Designer retrieves values from external sources.
- com.informatica.b2b.dx.client.tpm.rmi. Contains classes, interfaces, and methods to use when you define the external sources from which to retrieve the list of values in the Forms Designer.

To use the API, you include JAR file dependencies in the classpath. Use the files from the following locations:

- <DXInstallationDir>\powercenter\lib\dx-client-powercenter-9.5.1.jar
- <DXInstallationDir>\powercenter\lib\log4j-1.2.16.jar

For more information about the B2B Data Exchange Forms Designer Java API, see the online API reference located in the following directory: <DXInstallationDir>\powercenter\javadoc\tpm\apidocs\index.html.
The following table lists the methods in the B2B Data Exchange Forms Designer Java API:

<table>
<thead>
<tr>
<th>Method</th>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>updateAccountCustomAttribute (String attributeId, List&lt;String&gt; allowedValues)</td>
<td>void</td>
<td>Replaces the list of values for a custom account attribute with values from an external list without setting a default value.</td>
</tr>
<tr>
<td>updateAccountCustomAttribute (String attributeId, List&lt;String&gt; allowedValues, String defaultValue)</td>
<td>void</td>
<td>Replaces the list of values for a custom account attribute with values from an external list and sets a default value.</td>
</tr>
<tr>
<td>updatePartnerCustomAttribute (String attributeId, List&lt;String&gt; allowedValues)</td>
<td>void</td>
<td>Replaces the list of values for a custom partner attribute with values from an external list without setting a default value.</td>
</tr>
<tr>
<td>updatePartnerCustomAttribute (String attributeId, List&lt;String&gt; allowedValues, String defaultValue)</td>
<td>void</td>
<td>Replaces the list of values for a custom partner attribute with values from an external list and sets a default value.</td>
</tr>
<tr>
<td>updateWorkflowParameters (String workflowName, String parameterId, List&lt;String&gt; allowedValues)</td>
<td>void</td>
<td>Replaces the list of values for a custom workflow parameter with values from an external list without setting a default value.</td>
</tr>
<tr>
<td>updateWorkflowParameters (String workflowName, String parameterId, List&lt;String&gt; allowedValues, String defaultValue)</td>
<td>void</td>
<td>Replaces the list of values for a custom workflow parameter with values from an external list and sets a default value.</td>
</tr>
</tbody>
</table>
INDEX

C

custom mappings
  developing 12, 13

D

Dashboard and reports
  adding custom panel 63
  creating custom report 63
  custom LogiXML properties 61
  customizing in Logi Info Studio 62
  default filters 58
  default JavaScript support files 60
  default KPIs 57
  default SQL constants 58
  default theme modifier 61
  deploying and testing in B2B Data Exchange 64
  installing Logi Info Studio 62
  KPIs definition 57
  overview 56
  setting up custom application in Logi Info Studio 63
  structure in Logi Info Studio 58
  developing custom mappings
  process 13

DX_Add_Document_To_Event
  definition 22
  ports 22
  properties 24

DX_Aggregate
  definition 24
  ports 25
  properties 26

DX_Complete_Correlation
  definition 26
  ports 26
  properties 27

DX_Correlation
  ports 27

DX_Create_Event
  definition 27
  ports 28
  properties 28

DX_Event_Attribute
  definition 29
  ports 29
  properties 30

DX_Event_Details
  definition 30
  ports 30
  properties 31

DX_Generate_Temporary_File
  definition 31
  ports 31
  properties 32

DX_Get_Document_File_Path (continued)
  ports 32
  properties 33

DX_Increment_Profile_Parameter
  definition 33
  ports 33
  properties 34

DX_Initiate_Correlation
  definition 34
  ports 34
  properties 35

DX_Profile_Parameters
  definition 35
  ports 36

DX_Release_Delayed_Events
  definition 37
  ports 37
  properties 37

DX_Resolve_Profile
  definition 37
  ports 38
  properties 38

DX_Throw_Error
  definition 39
  ports 39
  properties 40

E

email
  monitor notification 53
  email for monitor notification
  configuring 53
  event reconciliation
  overview 43

F

forms designer
  actions 49
  definition 48
  Designer tab 48
  element properties 50
  group properties 51
  using 51

Forms Designer
  Java API reference 99

M

monitor notification
  email 53
Pending reconciliation
reconciliation status 45
PowerCenter
create mapping 16
creating source definition 13
creating target definition 15
creating the workflow 17
export workflow 18
integration 12
JMS connection 16, 17
JNDI connection 16
sources and targets 13
PowerCenter integration
developing custom mappings 12

Reconciled after timeout
reconciliation status 45
reconciliation complete
reconciliation status 45
reconciliation status ranking 45
reconciled after timeout 45
reconciliation complete 45
reconciliation correlation 45
timed out 45
reconciliations
completing 44
definition 43
initiating 44

timed out
reconciliation status 45
Transformation
DX_Add_Document_To_Event 22
DX_Aggregate 24
DX_Complete_Correlation 26
DX_Create_Event 27
DX_Event_Attribute 29
DX_Event_Details 30
DX_Generate_Temporary_File 31
DX_Get_Document_File_Path 32
DX_Increment_Profile_Parameter 33
DX_Initiate_Correlation 34
DX_Profile_Parameters 35
DX_Release_Delayed_Events 37
DXResolve_Profile 37
DX_Throw_Error 39
Transformations
configuring 21
definition 19
error handling 21
guidelines 21
installing plug-ins 20
Workflow parameters
customizing 51
workflows
properties 41