Agreement Supplemental License Terms, the BSD License (http://www.opensource.org/licenses/bsd-license.php) and the MIT License (http://www.opensource.org/licenses/mit-license.php).

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 Software is protected by U.S. Patent Numbers 5,794,246; 6,014,670; 6,016,501; 6,029,178; 6,032,158; 6,035,307; 6,044,374; 6,092,086; 6,208,990; 6,339,775; 6,640,226; 6,789,096; 6,820,077; 6,823,373; 6,850,947; 6,895,471; 7,117,215; 7,162,843; 7,254,590; 7,281,001; 7,421,458; 7,496,588; 7,523,121; 7,584,422; 7,720,842; 7,721,270; and 7,774,791, international Patents and other Patents Pending.

DISCLAIMER: Informatica Corporation 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 Corporation 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-91000-0001
Table of Contents

Preface ........................................................................... v
Informatica Resources. ...................................................... v
  Informatica Customer Portal. ........................................... v
  Informatica Documentation. ........................................... v
  Informatica Web Site. ...................................................... v
  Informatica How-To Library. ........................................... v
  Informatica Knowledge Base. ........................................... vi
  Informatica Multimedia Knowledge Base. ....................... vi
  Informatica Global Customer Support. ............................ vi

Chapter 1: Using PowerCenter with B2B Data Exchange. ......................... 1
  Overview of Using PowerCenter with B2B Data Exchange. .................. 1
  Sample Scenarios. ................................................................ 2
  Step 1. Install the B2B Data Exchange Plug-Ins. .......................... 2
    Server Plug-In. ................................................................ 2
    Client Plug-In. .................................................................. 2
  Step 2. Create the Source and Target Definitions. .......................... 2
    Source Definition. ................................................................ 3
    Target Definition. ................................................................ 4
  Step 3. Create the Mapping. .................................................. 5
    Workflow Parameters. ...................................................... 5
    Transformations. .......................................................... 5
  Step 4. Create the Connection Objects. .................................... 5
    JNDI Connection. .......................................................... 5
    JMS Connection. .......................................................... 6
  Step 5. Create the Workflow. .................................................. 6
  Step 6. Export the PowerCenter Workflow and Mapping. ................... 8
    Updating the Workflow. .................................................... 8

Chapter 2: B2B Data Exchange Transformations.............................................. 9
  B2B Data Exchange Transformations Overview. .......................... 9
  Installation and Registration. .............................................. 10
  Guidelines for Using the Data Exchange Transformations. ............... 10
  Configuring Data Exchange Transformations. ............................. 11
  Error Handling. ............................................................ 11
  DX_Add_Document_To_Event. ............................................ 12
    Input Ports. ................................................................. 12
    Input/Output Ports. .................................................... 12
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Exchange Properties.</td>
<td>13</td>
</tr>
<tr>
<td><strong>DX_Aggregate</strong></td>
<td>13</td>
</tr>
<tr>
<td>Input Ports</td>
<td>14</td>
</tr>
<tr>
<td>Input/Output Ports</td>
<td>14</td>
</tr>
<tr>
<td>Data Exchange Properties.</td>
<td>15</td>
</tr>
<tr>
<td><strong>DX_Complete_Correlation</strong></td>
<td>15</td>
</tr>
<tr>
<td>Input Ports</td>
<td>15</td>
</tr>
<tr>
<td>Input/Output Ports</td>
<td>15</td>
</tr>
<tr>
<td>Data Exchange Properties.</td>
<td>16</td>
</tr>
<tr>
<td><strong>DX_Create_Event</strong></td>
<td>16</td>
</tr>
<tr>
<td>Input Ports</td>
<td>16</td>
</tr>
<tr>
<td>Input/Output Ports</td>
<td>16</td>
</tr>
<tr>
<td>Data Exchange Properties.</td>
<td>17</td>
</tr>
<tr>
<td><strong>DX_Event_Attribute</strong></td>
<td>17</td>
</tr>
<tr>
<td>Input/Output Ports</td>
<td>18</td>
</tr>
<tr>
<td>Data Exchange Properties.</td>
<td>18</td>
</tr>
<tr>
<td><strong>DX_Event_Details</strong></td>
<td>18</td>
</tr>
<tr>
<td>Input/Output Ports</td>
<td>18</td>
</tr>
<tr>
<td>Data Exchange Properties.</td>
<td>19</td>
</tr>
<tr>
<td><strong>DX_Generate_Temporary_File</strong></td>
<td>19</td>
</tr>
<tr>
<td>Input/Output Ports</td>
<td>20</td>
</tr>
<tr>
<td>Data Exchange Properties.</td>
<td>20</td>
</tr>
<tr>
<td><strong>DX_Get_Document_File</strong></td>
<td>20</td>
</tr>
<tr>
<td>Input/Output Ports</td>
<td>20</td>
</tr>
<tr>
<td>Data Exchange Properties.</td>
<td>21</td>
</tr>
<tr>
<td><strong>DX_Increment_Profile_Parameter</strong></td>
<td>21</td>
</tr>
<tr>
<td>Input/Output Ports</td>
<td>21</td>
</tr>
<tr>
<td>Data Exchange Properties.</td>
<td>21</td>
</tr>
<tr>
<td><strong>DX_Initiate_Correlation</strong></td>
<td>21</td>
</tr>
<tr>
<td>Input Ports</td>
<td>22</td>
</tr>
<tr>
<td>Input/Output Ports</td>
<td>22</td>
</tr>
<tr>
<td>Data Exchange Properties.</td>
<td>22</td>
</tr>
<tr>
<td><strong>DX_Profile_Parameters</strong></td>
<td>22</td>
</tr>
<tr>
<td>Input/Output Ports</td>
<td>23</td>
</tr>
<tr>
<td>Data Exchange Properties.</td>
<td>23</td>
</tr>
<tr>
<td><strong>DX_Release_Delayed_Events</strong></td>
<td>23</td>
</tr>
<tr>
<td>Input/Output Ports</td>
<td>24</td>
</tr>
<tr>
<td>Data Exchange Properties.</td>
<td>24</td>
</tr>
<tr>
<td><strong>DX_Resolve_Profile</strong></td>
<td>24</td>
</tr>
<tr>
<td>Input Ports</td>
<td>24</td>
</tr>
<tr>
<td>Input/Output Ports</td>
<td>25</td>
</tr>
<tr>
<td>Data Exchange Properties.</td>
<td>25</td>
</tr>
<tr>
<td><strong>DX_Throw_Error</strong></td>
<td>25</td>
</tr>
<tr>
<td>Input Ports</td>
<td>26</td>
</tr>
</tbody>
</table>

ii Table of Contents
Input/Output Ports ..................................................... 26
Data Exchange Properties ................................................ 26

**Chapter 3: Creating Workflows for Event Reconciliation** .................. 27
Overview of Event Reconciliation .............................................. 27
Initiating and Completing an Event Reconciliation ......................... 28
  - Initiating an Event Reconciliation ......................................... 28
  - Completing an Event Reconciliation ........................................ 28
Reconciliation Status ....................................................... 29

**Chapter 4: Forms Designer** ..................................................... 31
Overview of the Forms Designer ............................................... 31
Using the Forms Designer ................................................... 32
Creating a Dependency between Fields ...................................... 35

**Chapter 5: Configuring Email for Monitor Notification** .................. 36
Overview of Email for Monitor Notification .................................. 36
The Built-in Email Notification Message Format .......................... 36
Configuring the Email Notifications ......................................... 37
  - Configuring Email Notifications Using System Properties ............... 37
  - Configuring Email Notifications Using Event Properties ............... 37
  - Configuring the Body of an Email Notification ................................ 38

**Chapter 6: B2B Data Exchange Web Services API** ........................ 39
Data Exchange Web Services Overview ...................................... 39
DX_TPM_Partner Web Service ................................................ 40
  - DX_TPM_Partner Request .................................................. 40
  - DX_TPM_Partner Response ................................................ 40
  - DX_TPM_Partner Create Operation ....................................... 41
  - DX_TPM_Partner Search Operation ...................................... 42
  - DX_TPM_Partner Update Operation ...................................... 43
  - DX_TPM_Partner Delete Operation ...................................... 44
  - DX_TPM_Partner Count Operation ...................................... 45
  - DX_TPM_Partner Get Operation ......................................... 46
DX_TPM_Account Web Service ................................................ 47
  - DX_TPM_Account Request ................................................ 47
  - DX_TPM_Account Response .............................................. 47
  - DX_TPM_Account Create Operation ...................................... 48
  - DX_TPM_Account Search Operation ...................................... 49
  - DX_TPM_Account Update Operation ...................................... 50
  - DX_TPM_Account Delete Operation ...................................... 51
  - DX_TPM_Account Count Operation ...................................... 51
  - DX_TPM_Account Get Operation ......................................... 52
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DX_TPM_Profile Web Service</td>
<td>53</td>
</tr>
<tr>
<td>DX_TPM_Profile Request</td>
<td>53</td>
</tr>
<tr>
<td>DX_TPM_Profile Response</td>
<td>54</td>
</tr>
<tr>
<td>DX_TPM_Profile Create Operation</td>
<td>54</td>
</tr>
<tr>
<td>DX_TPM_Profile Search Operation</td>
<td>56</td>
</tr>
<tr>
<td>DX_TPM_Profile Update Operation</td>
<td>56</td>
</tr>
<tr>
<td>DX_TPM_Profile Delete Operation</td>
<td>58</td>
</tr>
<tr>
<td>DX_TPM_Profile Count Operation</td>
<td>58</td>
</tr>
<tr>
<td>DX_TPM_Profile Get Operation</td>
<td>59</td>
</tr>
<tr>
<td>DX_Endpoint Web Service</td>
<td>60</td>
</tr>
<tr>
<td>DX_Endpoint Request</td>
<td>60</td>
</tr>
<tr>
<td>DX_Endpoint Response</td>
<td>60</td>
</tr>
<tr>
<td>Complex Type Objects in the Data Exchange Web Services</td>
<td>61</td>
</tr>
<tr>
<td>DXPartner Object</td>
<td>61</td>
</tr>
<tr>
<td>DXAccount Object</td>
<td>62</td>
</tr>
<tr>
<td>DXProfile Object</td>
<td>62</td>
</tr>
<tr>
<td>DXContactInformation Object</td>
<td>63</td>
</tr>
<tr>
<td>General B2B Data Exchange Web Service Behavior</td>
<td>63</td>
</tr>
<tr>
<td>B2B Data Exchange Web Service Security</td>
<td>64</td>
</tr>
<tr>
<td>User Authentication</td>
<td>64</td>
</tr>
<tr>
<td>User Access</td>
<td>65</td>
</tr>
<tr>
<td>B2B Data Exchange Web Services Setup</td>
<td>65</td>
</tr>
<tr>
<td>Web Service Import to PowerCenter</td>
<td>66</td>
</tr>
<tr>
<td>Appendix A: B2B Data Exchange Java API</td>
<td>67</td>
</tr>
<tr>
<td>Overview of B2B Data Exchange API</td>
<td>67</td>
</tr>
<tr>
<td>B2B Data Exchange API Versions</td>
<td>67</td>
</tr>
<tr>
<td>API Reference</td>
<td>68</td>
</tr>
<tr>
<td>Index</td>
<td>71</td>
</tr>
</tbody>
</table>
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.

Informatica Resources

Informatica Customer Portal

As an Informatica customer, you can access the Informatica Customer Portal site at http://mysupport.informatica.com. The site contains product information, user group information, newsletters, access to the Informatica customer support case management system (ATLAS), the Informatica How-To Library, the Informatica Knowledge Base, the Informatica Multimedia Knowledge Base, Informatica Product Documentation, and access to the Informatica user community.

Informatica Documentation

The Informatica Documentation team takes 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 informatica_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 http://mysupport.informatica.com.

Informatica Web Site

You can access the Informatica corporate web site at http://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 http://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 http://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 Multimedia Knowledge Base

As an Informatica customer, you can access the Informatica Multimedia Knowledge Base at http://mysupport.informatica.com. The Multimedia Knowledge Base is a collection of instructional multimedia files that help you learn about common concepts and guide you through performing specific tasks. If you have questions, comments, or ideas about the Multimedia Knowledge Base, contact the Informatica Knowledge Base team through email at KB_Feedback@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.

Use the following telephone numbers to contact Informatica Global Customer Support:

<table>
<thead>
<tr>
<th>North America / South America</th>
<th>Europe / Middle East / Africa</th>
<th>Asia / Australia</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Toll Free</strong></td>
<td><strong>Toll Free</strong></td>
<td><strong>Toll Free</strong></td>
</tr>
<tr>
<td>Brazil: 0800 891 0202</td>
<td>France: 00800 4632 4357</td>
<td>Australia: 1 800 151 830</td>
</tr>
<tr>
<td>Mexico: 001 888 209 8853</td>
<td>Germany: 00800 4632 4357</td>
<td>New Zealand: 1 800 151 830</td>
</tr>
<tr>
<td>North America: +1 877 463 2435</td>
<td>Israel: 00800 4632 4357</td>
<td>Singapore: 001 800 4632 4357</td>
</tr>
<tr>
<td><strong>Standard Rate</strong></td>
<td>Italy: 800 915 985</td>
<td><strong>Standard Rate</strong></td>
</tr>
<tr>
<td>North America: +1 650 653 6332</td>
<td>Netherlands: 00800 4632 4357</td>
<td>India: +91 80 4112 5738</td>
</tr>
<tr>
<td></td>
<td>Portugal: 800 208 360</td>
<td><strong>Standard Rate</strong></td>
</tr>
<tr>
<td></td>
<td>Spain: 900 813 166</td>
<td>Germany: 01805 702702</td>
</tr>
<tr>
<td></td>
<td>Switzerland: 00800 4632 4357</td>
<td><strong>Standard Rate</strong></td>
</tr>
<tr>
<td></td>
<td>or 0800 463 200</td>
<td>Netherlands: 030 6022 797</td>
</tr>
<tr>
<td></td>
<td>United Kingdom: 00800 4632 4357 or 0800 023 4632</td>
<td></td>
</tr>
</tbody>
</table>
Using PowerCenter with B2B Data Exchange

This chapter includes the following topics:

- Overview of Using PowerCenter with B2B Data Exchange, 1
- Step 1. Install the B2B Data Exchange Plug-Ins, 2
- Step 2. Create the Source and Target Definitions, 2
- Step 3. Create the Mapping, 5
- Step 4. Create the Connection Objects, 5
- Step 5. Create the Workflow, 6
- Step 6. Export the PowerCenter Workflow and Mapping, 8

Overview of Using PowerCenter with B2B Data Exchange

Use the PowerCenter Client tools to develop workflows to process documents for B2B Data Exchange. You create the mappings and workflows for B2B Data Exchange in the same way you create other PowerCenter workflows. B2B Data Exchange provides transformations specifically designed to perform the transformation tasks required to process Data Exchange documents. When you add the transformations to a mapping, you can set properties that are used by the corresponding workflow in B2B Data Exchange.

To develop a workflow in PowerCenter to process Data Exchange documents, complete the following steps:

1. Install and register the Data Exchange plug-in in a PowerCenter repository.
2. Create source (JMS, flat file, Data Transformation) and target definitions.
3. Create a mapping and use the B2B Data Exchange transformations, Unstructured Data transformations, and other PowerCenter transformations to process the documents.
4. Create the connection objects for the JMS queues.
5. Create the workflow and add the session to run the mapping.
6. Export the PowerCenter workflow, including mapping, to use as a workflow definition file in B2B Data Exchange.
Sample Scenarios

Sample scenarios are located in the `<DX Installation dir>/powercenter/samples` directory. What the samples do and how to use them are documented in the associated _readmefirst.txt_ files.

Step 1. Install the B2B Data Exchange Plug-Ins

Use the B2B Data Exchange installer to install the Data Exchange server and client plug-ins.

Server Plug-In

When you install the Data Exchange server plug-in, the installer copies the server plug-in file named `dxplugin.xml` to the following directory:

```
<DXInstallationDir>/powercenter/plugin<VERSION>/dxplugin.xml
```

Use Informatica Administrator (the Administrator tool) to register the plug-in to the repository where you plan to build mappings and workflows for B2B Data Exchange. The Data Exchange transformations in PowerCenter are disabled until you register the server plug-in. For more information about registering the plug-in, see the _Informatica Administrator Guide_.

Client Plug-In

When you install the Data Exchange client plug-in, the installer registers the plug-in file with the PowerCenter Client tools.

Step 2. Create the Source and Target Definitions

B2B Data Exchange uses JMS to send and receive documents to and from PowerCenter real-time workflows. The documents you send to JMS can be text or binary.

When using a real-time workflow, you must add a JMS source definition to a mapping designed to receive and process Data Exchange documents. If you send documents to B2B Data Exchange after processing, you must also define a JMS target. You create the JMS source and target definitions for Data Exchange mappings in the PowerCenter Designer in the same way you create JMS source and target definitions for other mappings.

PowerCenter batch workflows can have a flat file source, or no source at all. For more information, see the _PowerCenter Designer Guide_.

The JMS source and target definitions require additional properties configured to work with B2B Data Exchange. These properties allow B2B Data Exchange to pass parameters from B2B Data Exchange to PowerCenter and from PowerCenter to B2B Data Exchange.

Batch workflows require additional mapping parameters to interact with B2B Data Exchange. These mapping parameters allow B2B Data Exchange to exchange information with PowerCenter. Batch workflows are invoked using the Web Services Hub.
Source Definition

After you create the JMS source, you can add ports to the JMS source to store Data Exchange properties needed to run the workflow.

To add ports to the JMS source, edit the source definition and add properties in the JMS Message Property Columns tab. The Designer adds the property to the list of ports in the JMS source. By default, the Designer adds the prefix Prop_ to the port name.

Use the PowerCenter Designer to define the batch workflow mapping parameters for the PowerCenter mapping.

B2B Data Exchange Properties

You can use the following B2B Data Exchange properties as JMS source properties in realtime workflows, or as mapping parameters in batch workflows, to have the Data Exchange Server pass this information to the workflows:

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accountName</td>
<td>The Data Exchange Server sets the port to the account name associated with the event to be processed by the workflow.</td>
</tr>
</tbody>
</table>
| accountNumber | The Data Exchange Server sets the port to the account number associated with the event to be processed by the workflow.  
**Note:** accountNumber is a string datatype, not a number. |
| application    | The Data Exchange Server sets the port to the application name associated with the event to be processed by the workflow. |
| dataByReference| The Data Exchange Server sets this property to True when the JMS message sent to PowerCenter contains a document reference. The Data Exchange Server sets it to False when the JMS message sent to PowerCenter contains the document data. |
| eventId        | The Data Exchange Server sets this property to the event Id of the related sent message. |
| isEventReprocessed | If the event was reprocessed, the Data Exchange Server sets this port to True. |
| partnerName    | The Data Exchange Server sets the port to the partner name associated with the event to be processed by the workflow.  
**Note:** accountName, accountNumber and partnerName can be null if you have a global profile. |
| profileId      | The Data Exchange Server sets the port to the profile Id associated with the event to be processed by the workflow. |
| profileName    | The Data Exchange Server sets the port to the profile name associated with the event to be processed by the workflow. |

If an event monitor executed your profile, you will have additional properties, in addition to those shown above.

For a regular delivery monitor, B2B Data Exchange adds the following properties:

- DXSentEventId. Contains the list of event IDs.
- DXMonitorId. The monitor ID.
- DXMonitorName. The monitor name.
- DXUser. The recipient users.
For a batch delivery monitor, B2B Data Exchange adds the following properties:

- **DXSentEventIds**: Contains the list of event IDs.
- **DXMonitorId**: The monitor ID.
- **DXMonitorName**: The monitor name.

### Target Definition

After you create the JMS target, edit the target definition and add Data Exchange properties in the JMS Message Property Columns tab. The Designer adds the property to the list of ports in the JMS target. By default, the Designer adds the prefix `Prop_` to the port name.

Add the following properties to the JMS target definition:

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXDataByReference</td>
<td>Required. Set this property to True to indicate that the JMS message sent back to B2B Data Exchange contains a document reference. Set it to False to indicate that the JMS message sent back to B2B Data Exchange contains the document data.</td>
</tr>
<tr>
<td>DXEventId</td>
<td>The Data Exchange Server uses this property to determine the event ID to use for the event in the next workflow. If the JMS target contains this property, the Data Exchange Server does not generate a new event ID for the next workflow. The Data Exchange Server uses the value of this property as the event ID for the next workflow.</td>
</tr>
<tr>
<td>DXParentEventId</td>
<td>The Data Exchange Server uses this property to determine the parent event ID to use for the event in the next workflow. If the JMS target contains this property, the Data Exchange Server generates a new event ID for the next workflow and uses the value of this property as the parent ID of the event ID of the next workflow.</td>
</tr>
<tr>
<td>DXStatusName</td>
<td>This property contains the event status name to which B2B Data Exchange should change the event status after successful event transformation.</td>
</tr>
<tr>
<td>endpointName</td>
<td>The Data Exchange Server uses this property to determine the endpoint to route to. If the property is set to an endpoint name, then this endpoint is used by the server to route the message.</td>
</tr>
<tr>
<td>sendToAccount</td>
<td>Used by the Data Exchange Server to determine which endpoint to route to. If the flag is True, the endpoint defined for the Partner/Account is used.</td>
</tr>
</tbody>
</table>

You can also add properties to the JMS target to store other Data Exchange values that you want to send back to B2B Data Exchange.

**Note**: Create a child event for each target if you need to send an output message to more than one target. When you create a child event for each target, you can follow the transferring status of each message separately. Set the DXParentId property in the JMS target for this purpose. Otherwise, all targets overwrite the same event status field and you cannot track the delivery status of any target.

### Target Routing Procedure

B2B Data Exchange determines the Profile or Endpoint to which send a message is sent, based on the following rules:

1. **endpointName**: If the endpointName is set, the message is sent to the specified endpoint.
2. **profileId/profileName**: If the profileId or profileName is set, the message is sent to the indicated profile.
3. **accountNumber/partnerName**:
If sendToAccount is set to True, then send the message to the endpoint associated with the given accountNumber/partnerName, which must be unique for the specified accountNumber/partnerName.

- OR-

If sendToAccount is not set (or set to False), then the profile is determined by the application, account, and partner.

**Note:** If you want to route to an endpoint instead of a profile, it is strongly recommended that you use the accountNumber/partnerName routing method with sendToAccount set to True.

---

**Step 3. Create the Mapping**

Create a mapping to process B2B Data Exchange documents in the same way you build other PowerCenter mappings. Add the JMS source and target definitions and the transformations you require.

**Workflow Parameters**

Use the DX_Profile_Parameters transformation in the mapping to get profile parameter values at runtime.

**Transformations**

Use Data Exchange transformations and Unstructured Data transformations to add specific Data Exchange functionality to the mapping. You can also add other PowerCenter transformations to the mapping.

---

**Step 4. Create the Connection Objects**

Before you create the connection objects for the Data Exchange workflows, verify that PowerExchange for JMS is installed on PowerCenter and you can create JMS connection objects.

JMS sources and targets require the following connection objects:

- JNDI application connection
- JMS application connection

**JNDI Connection**

Create a JNDI application connection object for the B2B Data Exchange JMS server. Set the attributes required for B2B Data Exchange.

The following table describes the attributes of the JNDI connection object that you must 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 specified for the B2B Data Exchange JMS provider. Set this attribute to the following value:</td>
</tr>
</tbody>
</table>
### JMS Connection

Create one JMS application connection object for the input queue of each JMS source in each Data Exchange workflow. The input queue configuration must match the workflow name in B2B Data Exchange that represents the PowerCenter workflow.

Create one JMS application connection object for the B2B Data Exchange JMS server. All JMS targets in Data Exchange workflows use the same JMS application connection object for the output queue. The configuration of the output queue can use the configuration for the default inbound JMS endpoint defined in the B2B Data Exchange configuration file.

The following table describes the attributes of the JMS connection object that you must 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 this attribute to QUEUE.</td>
</tr>
<tr>
<td>JMS Connection Factory Name</td>
<td>Name of the connection factory in the JMS provider. Set this attribute to the following value: connectionfactory.local</td>
</tr>
</tbody>
</table>
| JMS Destination            | Name of the destination. The destination name must have the following format: queue,<DXWorkflowName>  
DXWorkflowName is the name of the workflow in B2B Data Exchange that represents the PowerCenter workflow. |

### Step 5. Create the Workflow

Create a workflow and add a session to run the mapping. You can create the workflow in the same way you create other PowerCenter workflows.

A JMS source is a real-time source. Configure the sessions in workflows that process Data Exchange documents as real-time sessions. Set the real-time properties, such as flush latency and commit type, to optimize processing.

To configure the real-time session in a Data Exchange workflow:

1. In the Workflow Designer, edit the session object.
2. In the Edit Tasks window, 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. Click the Mapping tab.
5. In the Sources section on the left pane, select the JMS source.
6. In the Properties section on the right pane, 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>

7. Save the session object.

For more information about real-time sessions, see the *PowerCenter Advanced Workflow Guide*.

Test the workflow to ensure that it works correctly and processes documents as expected.

**Note:** When you create a real time workflow, the source body of the JMS message can be formatted in either text or binary format. The appropriate TextMessage values are `TextMessage` or `BytesMessage` format. MapMessage is not supported.

The following figure shows how to configure the message format:
Step 6. Export the PowerCenter Workflow and Mapping

After you test the workflow and confirm that it is working correctly, export it. For batch workflows, you must export the workflow from the Repository Manager. For real-time workflows, you can export either the mapping or the workflow.

You create a B2B Data Exchange workflow in the Operation Console to represent the workflow in PowerCenter. When you create the Data Exchange workflow, use the export file as the definition file.

You can use the Forms Designer to design the layout of the workflow parameters for the operator. For more information about the Forms Designer, see “Using the Forms Designer” on page 32.

Updating the Workflow

If you make changes to the workflow, export it again and update the definition file in the B2B Data Exchange Operation Console.

Whenever a workflow is imported, B2B Data Exchange displays a list of its dependent profiles. Resolve changes to added/deleted workflow parameters, as well as parameters with changed types. When a workflow parameter type is changed, the import mechanism clears imported parameters. In this case, re-enter parameter values.

You can disable dependent profiles until they are reviewed by the operator.
B2B Data Exchange Transformations

This chapter includes the following topics:

- B2B Data Exchange Transformations Overview, 9
- DX_Add_Document_To_Event, 12
- DX_Aggregate, 13
- DX_Complete_Correlation, 15
- DX_Create_Event, 16
- DX_Event_Attribute, 17
- DX_Event_Details, 18
- DX_Generate_Temporary_File, 19
- DX_Get_Document_File_Path, 20
- DX_Increment_Profile_Parameter, 21
- DX_Initiate_Correlation, 21
- DX_PROFILE_Parameters, 22
- DX_Release_Delayed_Events, 23
- DX_Resolve_Profile, 24
- DX_Throw_Error, 25

B2B Data Exchange Transformations Overview

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

The Data Exchange transformations are custom Java transformations that allow you to access the B2B Data Exchange API without having to write code. You can use these transformations, the Unstructured Data transformation, and other transformations to process the B2B Data Exchange documents. If you require other functions that are not included in the transformations provided by B2B Data Exchange, you can create additional custom transformations and use the methods in the B2B Data Exchange 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_Event_Attribute</td>
<td>Gets or sets the value of event attributes.</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_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 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_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>

Installation and Registration

To use the Data Exchange transformations, run the B2B Data Exchange installer to install the following components:

- **Data Exchange server plug-in for PowerCenter.** Use the Informatica Administrator Tool to register the server plug-in.
- **Data Exchange client plug-in for PowerCenter.** You can use the B2B Data Exchange installer to install and register the client plug-in.

If the Data Exchange transformations are disabled in the PowerCenter Designer, the plug-ins are not properly registered. Verify that you have installed and registered the plug-ins correctly. For more information, see the *B2B Data Exchange Installation and Configuration Guide*.

Guidelines for Using the Data Exchange Transformations

Use the following guidelines when you work with the Data Exchange transformations:

- The Data Exchange transformations are based on the PowerCenter Custom transformation. They provide the same configuration options as other custom transformations. You can use them as you use other PowerCenter transformations.
- All Data Exchange transformations are connected transformations.
- All Data Exchange transformations are passive except the DX_Aggregate transformation.
If a port in the transformation has a corresponding Data Exchange property, the value of the port takes precedence over the value of the property at runtime. When the session runs, if the value of the port is not null, the Integration Service uses the value of the port for processing. If the value of the port is null, the Integration Service uses the value of the Data Exchange property for processing.

- Port names are case insensitive and prefix insensitive. DXEventID, dxEVENTid, and eventid are all the same.

When running 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 types to be used in the transformation. An error message indicates a failed connection. In this situation, PowerCenter gets event type values from the client plug-in configuration file.

If the connection fails, you might get an error message indicating that connecting to the B2B Data Exchange repository failed, and the values of the event type selections are taken from the client plug-in configuration file.

To resolve this issue, verify that the relevant section of the dxplugin.ini file is configured as follows:

```ini
[ODBC Connection String]
; ODBC connection string to the DX repository
; CONNECTION_STRING=DRIVER={DataDirect 6.1 Oracle Wire Protocol};UID=\1;PWD=\2;
  ; Host=localhost;Port=1521;DB=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
```

### Configuring Data Exchange Transformations

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

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

- Transformation tab. You can rename the transformation and add a description.
- Ports. You can add, edit, or delete ports or set ports to input or output.
- Properties tab. You can configure transformation properties such as module and function identifiers, transaction properties, and the runtime location. The properties displayed 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. You can set the default values for the ports in the transformation. You can also set the default values for other Data Exchange properties.

### Error Handling

Each Data Exchange transformation uses the following ports in error handling:

- DXErrorCode. When a transformation fails, the transformation sets the DXErrorCode to a value greater than zero.
- DXErrorMessage. When a transformation fails, the transformation stores an error message in the DXErrorMessage port to describe the failure.

When a transformation generates an error, the transformation performs the following tasks:

- 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 sets the status of the event to error.

**DX_Add_Document_To_Event**

This transformation attaches a document to an event. For example, you can use this transformation to attach a log to an event.

**Input Ports**

The DX_Add_Document_To_Event transformation has the following 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>DXEncoding</td>
<td>string</td>
<td>Character encoding of the document to attach to the event. This is the character set to use when converting 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 a new file. The DX_Add_Document_To_Event transformation saves the file as a new document reference in B2B Data Exchange and attaches the file 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**

The DX_Add_Document_To_Event transformation has the following 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. ID of the event to which a document will be attached.</td>
</tr>
<tr>
<td>DXData</td>
<td>string binary text</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, change 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 the value is false, the DXData port contains the document data.</td>
</tr>
<tr>
<td>DXDocumentId</td>
<td>string</td>
<td>ID 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>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value of the DXErrorCode port is greater than zero.</td>
</tr>
</tbody>
</table>

### Data Exchange Properties

You can configure the following Data Exchange properties in 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. Indicates how the</td>
</tr>
<tr>
<td></td>
<td>document is used in the event. Select one of the following roles:</td>
</tr>
<tr>
<td></td>
<td>- SOURCE. The document is attached to an event as a source document. When</td>
</tr>
<tr>
<td></td>
<td>you view the event on the B2B Data Exchange Operation Console, you can</td>
</tr>
<tr>
<td></td>
<td>view the attached document in the section labeled Input.</td>
</tr>
<tr>
<td></td>
<td>- TARGET. The document is attached to an event as a target document. When</td>
</tr>
<tr>
<td></td>
<td>you view the event on the B2B Data Exchange Operation Console, you can</td>
</tr>
<tr>
<td></td>
<td>view the attached document in the section labeled Output.</td>
</tr>
<tr>
<td></td>
<td>- LOG. The document is attached to an event as a log document. When you</td>
</tr>
<tr>
<td></td>
<td>view the event on the B2B Data Exchange Operation Console, you can view</td>
</tr>
<tr>
<td></td>
<td>the attached document in the section labeled Logging Information.</td>
</tr>
<tr>
<td>Generate an error in case a failure occurs in</td>
<td>Indicates whether to set the status of the event to ERROR when the</td>
</tr>
<tr>
<td>this transformation</td>
<td>transformation generates an error. Default is true.</td>
</tr>
</tbody>
</table>

### DX_Aggregate

This transformation groups segments of data together for processing. For example, you previously split a large document so the workflow processes it in row sets. You can use the DX_Aggregate transformation to aggregate the row sets back into a single document to send to a partner.

The data going into the DX_Aggregate transformation must be sorted by the DXParentEventId and DXGroupByKey ports. The transformation groups rows by the DXParentEventId and DXGroupByKey ports.

The DX_Aggregate transformation is an active transformation.

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. Then it 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. Then it passes the output to the pipeline.
4. If the DXData port for a row is null, the transformation does not pass the row to the pipeline.
5. The DX_Aggregate transformation drops rows that contain null values in the DXData port. It does not pass them out to the pipeline.

6. For pass-through ports, the transformation sends the value from the last row of the group to the pipeline. It does not pass error and null rows to the pipeline.

The DX_Aggregate transformation uses indicators to determine the end of the group. If the following indicators are true, the transformation has reached the end of the data to aggregate:

- End of data.
- 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 has reached the value in the DXMaxRowsInGroup port.
- The value in the DXLastInGroup port is true.

**Input Ports**

The DX_Aggregate transformation has the following 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>ID 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>Used with the parent event ID to group rows.</td>
</tr>
<tr>
<td>DXDelimiter</td>
<td>string</td>
<td>Delimiter string to add between segments of data that indicates the end of a segment and the start of another segment.</td>
</tr>
<tr>
<td>DXData</td>
<td>string, binary, text</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 the value is false, the DXData port contains data.</td>
</tr>
<tr>
<td>DXLastInGroup</td>
<td>string</td>
<td>Indicates that the row is the last row in a group.</td>
</tr>
<tr>
<td>DXMaxRowsInGroup</td>
<td>integer</td>
<td>Indicates the maximum number of rows in a group.</td>
</tr>
</tbody>
</table>

**Input/Output Ports**

The DX_Aggregate transformation has the following 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 has processed since the Integration Service was started. The transformation increments the value of this port every time a group is processed. When the 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>
</tbody>
</table>
### Port Description

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 B2B Data Exchange document.</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>
<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>

### Data Exchange Properties

You can configure the following Data Exchange properties in the DX_Aggregate 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. Default is true.</td>
</tr>
</tbody>
</table>

### DX_Complete_Correlation

This transformation completes the correlation for an event. The transformation requires the correlation ID generated when you used the DX_Initiate_Correlation transformation to initiate the correlation.

### Input Ports

The DX_Complete_Correlation transformation has the following 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. ID of the correlation to complete.</td>
</tr>
</tbody>
</table>

### 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. ID 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>
Data Exchange Properties

You can configure the following Data Exchange properties in 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 generate an error if the correlation ID does not exist. Default is true.</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. Default is true.</td>
</tr>
</tbody>
</table>

DX_Create_Event

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

Input Ports

The DX_Create_Event transformation has the following 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>ID of parent event. If the DXParentEventId port is not null, the transformation creates a child event of the DXParentEventId event. If the DXParentEventId port is null, and the property “Create root event” is selected, then the transformation creates a root event.</td>
</tr>
<tr>
<td>DXSubject</td>
<td>string</td>
<td>Subject of the event to create. Can be up to 255 bytes long.</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

The DX_Create_Event transformation has the following 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>Event ID for the generated event.</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>ID of the profile associated with the new event. Mandatory.</td>
</tr>
</tbody>
</table>
If the profile ID is not null, the transformation associates the profile ID with the event created by the transformation. If the profile ID is null, the transformation uses the DXApplication, DXAccountNumber, and DXPartnerName ports to determine the profile ID to associate with the event created by the transformation.

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>If the profile ID is not null, the transformation associates the profile ID with the event created by the transformation. If the profile ID is null, the transformation uses the DXApplication, DXAccountNumber, and DXPartnerName ports to determine the profile ID to associate with the event created by the transformation.</td>
</tr>
</tbody>
</table>

## Data Exchange Properties

You can configure the following Data Exchange properties in 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>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. Default is true.</td>
</tr>
</tbody>
</table>

## DX_Event_Attribute

You use the DX_Event_Attribute transformation to get or set event attribute values. To use this transformation, you must first configure an event attribute in the Operation Console, then use this transformation to set or get its value.

You can add ports to DX_Event_Attribute that represent the event attributes you want to get or set. The port name must match the event attribute name, but the attribute name is not case sensitive. For example, `event_att1` and `EVENT_ATT1` are treated as the attribute name. Note that unlike other profile parameters, the event name is not prefix insensitive. For example, `DX_event_att1` and `event_att1` are not the same. If you set a value for the port, the transformation sets the event attribute to this value. If the port is value is Null, then DX_Event_Attribute gets (retrieves) the value of the event attribute.

**Note:** The DXValue port, DXAttributeKey, and Event attribute name properties, used in previous versions of B2B Data Exchange, are optional. Use the `<attribute>` port instead.
Input/Output Ports

The DX_Event_Attribute transformation has the following 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. ID of the event associated with the attribute to get or 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>

Data Exchange Properties

You can configure the following Data Exchange properties in the DX_Event_Attribute transformation:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event attribute name</td>
<td>Event attribute to update. The property must be configured in the B2B Data Exchange Operation Console. If the value is null, the transformation generates an error. Note: This property is optional. It is recommended that you use the &lt;Attribute&gt; port instead. This property is retained for backward compatibility.</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. Default is true.</td>
</tr>
</tbody>
</table>

DX_Event_Details

This transformation gets or sets multiple properties of an event.

The ports represent the properties of an event. The transformation sets the property of the event with the value passed to the transformation through an input port. The transformation passes the value of the property through an output port.

Input/Output Ports

The DX_Event_Details transformation has the following input/output ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXStatusName</td>
<td>string</td>
<td>Status of the event. The DXStatusName property provides progress information on the various steps that the event passed through during its journey through the B2B Data Exchange processing cycle. B2B Data Exchange sets event status to Transferring when writing a message to a target. Status is changed to Complete or Error, depending on whether the transfer was successful, or not. If you want B2B Data Exchange to use a different status other than</td>
</tr>
</tbody>
</table>
Port | Type | Description
--- | --- | ---
--- | --- | ---
 DXEventId | string | Required. ID 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 | ID 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.

### Data Exchange Properties

You can configure the following Data Exchange properties in 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. B2B Data Exchange creates events with the default value of Event Subject &quot;in progress&quot;. We recommend that you change this property to a more meaningful value. You can use the DX_Event_Details transformation to update the Event Subject 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. Default is true.</td>
</tr>
</tbody>
</table>

### DX_Generate_Temporary_File

This transformation gets the path and file name for a document in the Data Exchange document store.

This transformation generates a unique file name in a temporary directory in the B2B Data Exchange document store. The transformation does not create a file. For example, you use an Unstructured Data transformation to write data to a temporary file. You can use this transformation to generate a file name for a file in the Data Exchange document store and write the data from the Unstructured Data transformation.
Input/Output Ports

The DX_Generate_Temporary_File transformation has the following 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. ID of the event to associate with the generated file path.</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>

Data Exchange Properties

You can configure the following Data Exchange properties in the DX_Generate_Temporary_File 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. Default is true.</td>
</tr>
</tbody>
</table>

DX_Get_Document_File_Path

This transformation determines the path and file name of a document based on a document reference.

Input/Output Ports

The DX_Get_Document_File_Path transformation has the following 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 document identified by a document reference.</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>
Data Exchange Properties

You can configure the following Data Exchange properties in 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. Default is true.</td>
</tr>
</tbody>
</table>

DX_Increment_Profile_Parameter

This transformation increments the value of a parameter associated with a profile. The parameter to be incremented must be numeric.

Input/Output Ports

The DX_Increment_Profile_Parameter transformation has the following 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>ID of the profile 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>

Data Exchange Properties

You can configure the following Data Exchange properties in 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. Default is true.</td>
</tr>
</tbody>
</table>

DX_Initiate_Correlation

This transformation initiates a reconciliation for an event. Use this transformation in a workflow that processes a document that requires reconciliation.
When a reconciliation is initiated, it needs to be completed. In a workflow, use the DX_Initiate_Correlation transformation to initiate a reconciliation. In another workflow, use the DX_Complete_Correlation transformation to complete the reconciliation. For more information, see “Initiating and Completing an Event Reconciliation” on page 28.

Input Ports

The DX_Initiate_Correlation transformation has the following 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>Description of the type of correlation to initiate. Required. The description can be a maximum of 255 bytes. Configure the correlation type as part of the transformation.</td>
</tr>
<tr>
<td>DXCorrelationId</td>
<td>string</td>
<td>ID of the correlation to be initiated. Required.</td>
</tr>
<tr>
<td>DXTimeWindow</td>
<td>integer</td>
<td>The time interval in seconds within which the correlation must be completed.</td>
</tr>
</tbody>
</table>

Input/Output Ports

The DX_Initiate_Correlation transformation has the following 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>ID of the event associated with the correlation to be initiated. Required.</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>

Data Exchange Properties

You can configure the following Data Exchange properties in 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>Generate error if the event is already associated with a correlation ID. Default is true.</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. Default is true.</td>
</tr>
</tbody>
</table>

DX_Profile_Parameters

You can use this transformation to get the parameter values defined for a profile.
Input/Output Ports

The DX_Get_Profile_Parameters transformation has the following 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. Profile ID.</td>
</tr>
<tr>
<td>DXEventId</td>
<td>string</td>
<td>Event ID. Optional. It is possible to provide EventId together with ProfileId. If EventId is provided, it looks up the parameter values from the Event Context. In case of exceptions, the associated event status will be set to Error.</td>
</tr>
<tr>
<td>DXParametersXML</td>
<td>string</td>
<td>Contains all the parameters defined for the profile, wrapped in an XML string that holds the parameter names and values. Each required parameter should have a port. In case of a value that represents multi-selection, it will be represented as a string containing a semicolon separated list of values. B2B Data Exchange generates the XML only if the output port is connected.</td>
</tr>
<tr>
<td>&lt;Parameter&gt;port</td>
<td>string</td>
<td>When you import a PowerCenter workflow or mapping files into B2B Data Exchange. B2B Data Exchange parses all occurrences of DX_Profile_Parameters transformations. All the ports in these transformations that are not B2B Data Exchange built-in ports, for example, eventid and profileid populate as profile parameters. Ports are identified by name. Ports with the same name are treated as a single parameter. If a parameter port does not match a defined parameter, B2B Data Exchange logs a debug message. All &lt;parameter&gt; ports are strings. You can change types in the B2B Data Exchange Operation Console using the Form Designer. Boolean: {true</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 DXError port is greater than zero.</td>
</tr>
</tbody>
</table>

DX_Release_Delayed_Events

This transformation releases the events delayed by a specific profile. Delayed events are released according to defined release rules. DX_Release_Delayed_Events overrides the release rules.
## Input/Output Ports

The DX_Release_Delayed_Events transformation has the following 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>ID of the profile associated with the delayed processing rule used to determine which events to release. Required if EventId list is null.</td>
</tr>
<tr>
<td>EventId list</td>
<td>string</td>
<td>EventId to release. Comma-separated list for multiple events. Required if DXProfileId is null.</td>
</tr>
</tbody>
</table>

## Data Exchange Properties

You can configure the following Data Exchange properties in 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>Release the event even if it is not within the scope of a 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. Default is false.</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. Default is true.</td>
</tr>
</tbody>
</table>

## DX_Resolve_Profile

This transformation determines the profile ID based on an application and a partner or account number. This transformation returns an error if the combination of an application and a partner or account number is not associated with any profile.

## Input Ports

The DX_Resolve_Profile transformation has the following 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 associated with the profile.</td>
</tr>
<tr>
<td>DXEventId</td>
<td>string</td>
<td>If the transformation fails, this event will be set to Error status.</td>
</tr>
</tbody>
</table>
Port | Type | Description
--- | --- | ---
DXPartnerName | string | Required. Name of the partner associated with the profile.
DXAccountNumber | string | The account number that is associated with the profile. If you use the account number and partner name to resolve the profile, then Account Number is required. If you use the profile name to resolve the profile, then the Account Number is not required.

### Input/Output Ports

The DX_Resolve_Profile transformation has the following 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>ID of the profile associated with the application and 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>

### Data Exchange Properties

You can configure the following Data Exchange properties in 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. Default is true.</td>
</tr>
</tbody>
</table>

### DX_Throw_Error

This transformation handles errors in the workflow. It generates an error when the transformation fails. It performs the following tasks:

- Sets the status of the associated event to ERROR.
- Creates the error message from value of the DXDescription port.
- Attaches the error message to the associated event.
- Logs the error in the session log.
Input Ports

The DX_Throw_Error transformation has the following 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. This is the error message added to the session log. This is also used as 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

The DX_Throw_Error transformation has the following 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. ID of the event associated with the error.</td>
</tr>
<tr>
<td>DXData</td>
<td>string</td>
<td>Log document to attach to the event. 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>
<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 the value is false, the DXData port contains the document data.</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>

Data Exchange Properties

You can configure the following Data Exchange properties in 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 for the error log document that this transformation attaches to the event.</td>
</tr>
<tr>
<td>Message type</td>
<td>Alphanumeric value to associate with the event. Any value is valid.</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. Default is true.</td>
</tr>
</tbody>
</table>
Creating Workflows for Event Reconciliation

This chapter includes the following topics:

- Overview of Event Reconciliation, 27
- Initiating and Completing an Event Reconciliation, 28
- Reconciliation Status, 29

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 of the B2B Data Exchange API to initiate the 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.

In the workflow to process the acknowledgment document, call the completeCorrelation method to complete the reconciliation. 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.

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 acknowledgment, 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” on page 21.

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 = "997";
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” on page 15.

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.receiverId + 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:

![Diagram showing reconciliation status process](image)

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

This chapter includes the following topics:

- Overview of the Forms Designer, 31
- Using the Forms Designer, 32
- Creating a Dependency between Fields, 35

Overview of the Forms Designer

The Forms Designer is a tool that developers and administrators can use to design the user interface of forms for various Operations Console screens that need to have a dynamic interface. These screens include:

- Partner > Additional Information
- Partner > Account > Additional Information
- Profile > Workflow Parameters

Use the Forms Designer to arrange the screen layout and to configure the individual fields. The Forms Designer lets you set up the following form properties:

- Labels
- Descriptions
- Mandatory flags
- Read-only flags
- Parameter types
- Parameter representations
- Default values
- Valid values
- Relationships between fields
- User interface screen groups, such as tabs, columns, and headings
Using the Forms Designer

To use the Forms Designer:

1. In the Navigator, click Partner Management > Workflows. Click the edit icon for the workflow you want to edit. B2B Data Exchange displays the Update Workflow screen.

2. Click Workflow Parameters.
   A list of workflow parameters displays. These are the workflows that were imported from the PowerCenter workflow.

3. Click Designer.
   The Forms Designer opens.

The following table explains the actions you can take on this screen:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit</td>
<td>Allows editing the properties of an element.</td>
</tr>
<tr>
<td>Create Tab</td>
<td>Creates a tab. A tab is a group of elements contained in a screen tab.</td>
</tr>
<tr>
<td>Create Column</td>
<td>Creates a column. A column is a group of elements that you can display beside another column, across the width of the form you are designing.</td>
</tr>
<tr>
<td>Create Header</td>
<td>Creates a header element. A header allows you to create a header above a group of parameters</td>
</tr>
<tr>
<td>Create Disclosure</td>
<td>Create a disclosure group. A disclosure group is a group of parameters that the operator can display or hide by clicking the name of the group.</td>
</tr>
<tr>
<td>Delete Group</td>
<td>Deletes a group. The elements in the deleted group remain in the workflow.</td>
</tr>
<tr>
<td>Move to Group</td>
<td>Move elements from one tab, column, header, or disclosure to another.</td>
</tr>
</tbody>
</table>

You can nest the following elements in your forms to a maximum of four levels:

- Tab
- Column
- Header
- Disclosure

You can nest tabs, columns, headers, and disclosures when you create groups, up to a maximum of four deep. For example, a hierarchy of four tabs might look like this:
The screen contains a grid display of the workflow elements. The following table describes the grid columns:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The unique identifier of the element.</td>
</tr>
<tr>
<td>Label</td>
<td>The label of the element, as it appears in the form.</td>
</tr>
<tr>
<td>Type</td>
<td>The datatype of the element, such as text or date.</td>
</tr>
<tr>
<td>Representation</td>
<td>The representation of the element in the form, for example, entry field, radio button, or drop-down list.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>If selected, the element is mandatory.</td>
</tr>
<tr>
<td>Read-Only</td>
<td>If selected, the element is read-only.</td>
</tr>
<tr>
<td>Depends On</td>
<td>If the element depends on another element, the name of the other element appears here.</td>
</tr>
</tbody>
</table>

4. Define the properties of each element, as described in the following table:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Yes</td>
<td>Name of the parameter in the form.</td>
</tr>
<tr>
<td>Description</td>
<td>No</td>
<td>Short description of the parameter. The description appears as a tool tip.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>No</td>
<td>Select if the parameter is mandatory in the form which force the operator to enter a value.</td>
</tr>
<tr>
<td>Read Only</td>
<td>No</td>
<td>Select if the parameter is read only in the form. Read only parameters require a default value.</td>
</tr>
<tr>
<td>Hidden</td>
<td>No</td>
<td>Hides a parameter from the Operator. Used for workflow parameters that are not available for Operator edit.</td>
</tr>
<tr>
<td>Type</td>
<td>No</td>
<td>Datatype of the parameter. Select from the list.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Mandatory</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Representation</td>
<td>No</td>
<td>The parameter's representation in the form. Select from the list.</td>
</tr>
<tr>
<td>Default Value</td>
<td>No</td>
<td>Initial value of the parameter in the form.</td>
</tr>
<tr>
<td>Values</td>
<td>No</td>
<td>A list of valid values for the parameter.</td>
</tr>
<tr>
<td>Minimum</td>
<td>No</td>
<td>Minimum value. Applies to numbers and string fields/types. If the field is a string, the attribute specifies the minimum length.</td>
</tr>
<tr>
<td>Maximum</td>
<td>No</td>
<td>Maximum value. Applies to numbers and string fields/types. If the field is a string, the attribute specifies the maximum length.</td>
</tr>
<tr>
<td>Enable/disable condition</td>
<td>No</td>
<td>For more information on enable/disable conditions, see “Creating a Dependency between Fields” on page 35</td>
</tr>
</tbody>
</table>

5. To create a tab on the form:
   a. Click Actions > Create Tab. Enter the tab properties and click OK. The tab properties are:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Yes</td>
<td>The tab label as it appears in the form.</td>
</tr>
<tr>
<td>Type</td>
<td>Yes</td>
<td>Tab.</td>
</tr>
<tr>
<td>Enable/Disable Condition</td>
<td>No</td>
<td>Allows you to enable or disable all the fields in the tab based on the value of another parameter in the form. To enable or disable a group, select the checkbox and fill in the parameters defining the enable/disable condition. For more information on enable/disable conditions, see “Creating a Dependency between Fields” on page 35</td>
</tr>
</tbody>
</table>

The tab appears in the workflow elements grid.

   b. Drag and drop the elements that should appear in the tab to the new tab element.

6. To arrange elements in a column, click Actions > Create Column and enter the properties for the column. Click Enter. Drag and drop the parameters to the new column. Create two or more columns to display them side by side.

   The column element appears in the element grid.

7. To create a header, click Actions > Create Header. Enter the header text in the Label field. Drag and drop the elements that should appear under the Header.

8. To create a disclosure group, click Actions > Create Disclosure. Enter a label for the disclosure and drag and drop elements to it. Define the enabled/disabled condition if required.

9. Click Save to save the workflow and the form.
The following figures shows the final design of the Workflow Parameters screen:

Creating a Dependency between Fields

The purpose of a dependency is to enable or disable one field or group of fields, based on the value of another field. For example, you might have a field that you want to display only if the value of the Choice Property field is "yes".

You can define a condition between two parameters, two groups, or a parameter and a group.

1. From the list of parameters, select the parameter or group that you want to enable or disable.
   B2B Data Exchange highlights the selected parameter.
2. Select **This group/parameter should be**.
3. Select **enabled** or **disabled** from the list.
4. Select **when** from the list.
5. Select **specific** or **any**.
6. If you chose specific in step 5, fill in the specific value.

The following figure illustrates the complete expression definition:

<table>
<thead>
<tr>
<th>This group/field should be</th>
<th>enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>when field</td>
<td>typeProperty</td>
</tr>
<tr>
<td>has</td>
<td>specific</td>
</tr>
<tr>
<td>value</td>
<td>Rental</td>
</tr>
</tbody>
</table>
Configuring Email for Monitor Notification

This chapter includes the following topics:

- Overview of Email for Monitor Notification, 36
- The Built-in Email Notification Message Format, 36
- Configuring the Email Notifications, 37

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.
You can use the built-in email format for event notifications, or you can customize it.

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><code>dx_email_bcc_field</code></td>
<td>A comma-separated list of recipient email addresses. These recipients are added to the addressees defined in the monitor.</td>
</tr>
<tr>
<td><code>dx_email_body_file</code></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><code>dx_email_body_field</code></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><code>dx_email_from_field</code></td>
<td>This string becomes the &quot;from field&quot; of the email.</td>
</tr>
</tbody>
</table>
### Event Property

<table>
<thead>
<tr>
<th>Event Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dx_email_cc_field</td>
<td>A comma-separated list of recipient email addresses. These recipients are</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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 comma-separated list of recipient email addresses. These recipients are</td>
</tr>
<tr>
<td></td>
<td>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.
This chapter includes the following topics:

- Data Exchange Web Services Overview, 39
- DX_TPM_Partner Web Service, 40
- DX_TPM_Account Web Service, 47
- DX_TPM_Profile Web Service, 53
- DX_Endpoint Web Service, 60
- Complex Type Objects in the Data Exchange Web Services, 61
- General B2B Data Exchange Web Service Behavior, 63
- B2B Data Exchange Web Service Security, 64
- B2B Data Exchange Web Services Setup, 65

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_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.

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 partner name identifies a system object.

**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:

```
  <soapenv:Header>
    <inf:Security>
      <sec:DXUsernameToken>
        <sec:DXUsername>sys</sec:DXUsername>
        <sec:DXPassword DXPasswordType="PasswordText">sys</sec:DXPassword>
        <![--sec:Nonces]]></sec:DXUsernameToken>
      </sec:DXSecurity>
    </inf:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tpm:DXPartnerRequest>
      <operation>create</operation>
    </tpm:DXPartnerRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
<partner>
  <Name>PartnerTestName</Name>
  <Description>PartnerTest Description</Description>
  <!--Zero or more repetitions-->
  <CategoryName>CategoryTest1</CategoryName>
  <!--Zero or more repetitions-->
  <CategoryName>CategoryTest2</CategoryName>
</partner>

<ContactInformation>
  <Name>PartnerTestContact1</Name>
  <Description>PartnerTestContact1 Description</Description>
  <Title>PartnerTestContact1 Title</Title>
  <Address>PartnerTestContact1 Address</Address>
  <PhoneNumber>12345678912</PhoneNumber>
  <BusinessPhoneNumber>12345678913</BusinessPhoneNumber>
  <FaxNumber>12345678914</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>12345678923</BusinessPhoneNumber>
  <FaxNumber>12345678924</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>
  <!--Zero or more repetitions-->
  <CategoryName>CategoryTest4</CategoryName>
  <!--Zero or more repetitions-->
  <AccountAttribute>
    <Name>AccountAttribute1</Name>
    <Value>AccountAttribute1Value</Value>
  </AccountAttribute>
  <AccountAttribute>
    <Name>AccountAttribute2</Name>
    <Value>AccountAttribute2Value</Value>
  </AccountAttribute>
</account>

</soapenv:Body>
</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 partner name identifies a system object.

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 partner name identifies a system object.

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:Header>
    <inf:Security>
      <UserNameToken>
        <Username>Administrator</Username>
        <Password>Administrator</Password>
      </UserNameToken>
    </inf:Security>
    <sec:DXSecurity>
      <sec:DXUsernameToken>
        <sys:DXUsername>sys</sys:DXUsername>
        <sys:DXPassword>DXPasswordType="PasswordText"></sys:DXPassword>
        <sec:NoncesFile></sec:NoncesFile>
      </sec:DXUsernameToken>
    </sec:DXSecurity>
  </soapenv:Header>
  <tpm:DXPartnerRequest>
    <operation>update</operation>
    <partner>
      <Name>PartnerTestName</Name>
      <Description>PartnerTest Description</Description>
      <CategoryName>CategoryTest1</CategoryName>
      <CategoryName>CategoryTest2</CategoryName>
      <ContactInformation>
        <Name>PartnerTestContact1</Name>
        <Description>PartnerTestContact1 Description</Description>
        <Title>PartnerTestContact1 Title</Title>
      </Address>PartnerTestContact1 Address</Address>
    </partner>
  </tpm:DXPartnerRequest>
</soapenv:Envelope>
```
When the operation is successful, the web service sends an empty response to the client.

The search string must match the partner name exactly. For example, `DX_TPM_Partner` does not match the partner name `Informatica`. The partner name is not case sensitive.

**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: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>
  <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: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>
  <tpm:DXPartnerRequest>
    <operation>count</operation>
  </tpm:DXPartnerRequest>
</soapenv:Body>
```
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: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>
        <![CDATA[<sec:Nonce>]]>
        <![CDATA[<sec:Created>]]>
      </sec:DXUsernameToken>
    </sec:DXSecurity>
  </soapenv:Header>
  <soapenv:Body>
    <tpm:DXPartnerRequest>
      <operation>get</operation>
      <index>0</index>
    </tpm:DXPartnerRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
DX_TPM_Account Web Service

The DX_TPM_Account web service to 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.

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:
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 system object.
- The partner for the account does not exist in the B2B Data Exchange repository.
- The partner name for the account identifies a system object.

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:Header>
    <inf:Security>
      <sec:UsernameToken>
        <Username>Administrator</Username>
        <Password>Administrator</Password>
      </sec:UsernameToken>
    </inf:Security>
  </soapenv:Header>
  <tpm:DXAccountRequest>
    <operation>create</operation>
  </tpm:DXAccountRequest>
</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 system object.

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:Header>
    <inf:Security>
      <sec:DXUsernameToken>
        <sec:DXUsername>sys</sec:DXUsername>
        <sec:DXPassword DXPasswordType="PasswordText">sys</sec:DXPassword>
      </sec:DXUsernameToken>
    </inf:Security>
  </soapenv:Header>
  <tpm:DXAccountRequest>
    <Name>AccountTestName</Name>
    <Number>123</Number>
    <Description>AccountTest Description</Description>
    <PartnerName>PartnerTestName</PartnerName>
    <AccountNumber>Test123</AccountNumber>
  </tpm:DXAccountRequest>
</soapenv:Envelope>
```
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 system object.

**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:Header>
    <inf:Security>
      <UserNameToken>
        <UserName>Administrator</UserName>
        <Password>Administrator</Password>
      </UserNameToken>
    </inf:Security>
    <sec:DXSecurity>
      <sec:DXUsernameToken>
        <sec:DXUsername>sys</sec:DXUsername>
        <sec:DXPasswordDXPasswordType="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:

```xml
<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: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 system object.

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:Header>
    <inf:Security>
      <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:DXAccountRequest>
      <operation>count</operation>
      <partnerName>myPartner</partnerName>
    </tpm:DXAccountRequest>
  </soapenv:Body>
</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 system object.

**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:Header>
    <inf:Security>
      <inf:UsernameToken>
        <inf:Username>Administrator</inf:Username>
        <inf:Password>Administrator</inf:Password>
      </inf:UsernameToken>
    </inf:Security>
  </soapenv:Header>
  <tpm:DXAccountRequest>
    <operation>get</operation>
    <partnerName>myPartner</partnerName>
    <index>0</index>
  </tpm:DXAccountRequest>
</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.

**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 system object.

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:

```xml
  <soapenv:Header>
    <inf:Security>
      <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>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>
        <!-- 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: -->
        <EventAttribute>
          <Name>EventAttribute1</Name>
          <Value>EventAttribute1Value</Value>
        </EventAttribute>
        <EventAttribute>
          <Name>EventAttribute2</Name>
          <Value>EventAttribute2Value</Value>
        </EventAttribute>
      </profile>
    </tpm:DXProfileRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
DX_TPM_Profile Search Operation

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 system object.

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:

```
  <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:DXProfileRequest>
      <operation>search</operation>
      <profileName>myProfile</profileName>
    </tpm:DXProfileRequest>
  </soapenv:Body>
</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 system object.

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:Header>
    <inf:Security>
      <UsernameToken>
        <Username>Administrator</Username>
        <Password>Administrator</Password>
      </UsernameToken>
    </inf:Security>
    <sec:DXSecurity>
      <sec:DXUserToken>
        <sec:DXUsername>sys</sec:DXUsername>
        <sec:DXPassword DXPasswordType="PasswordText">sys</sec:DXPassword>
      </sec:DXUserToken>
    </sec:DXSecurity>
  </soapenv:Header>
  <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:--> 
      <CategoryId>CategoryName1</CategoryId>
      <CategoryId>CategoryName2</CategoryId>
      <!-- Zero or more repetitions:--> 
      <EventAttribute>
        <Name>EventAttribute1</Name>
        <Value>EventAttribute1Value</Value>
      </EventAttribute>
      <EventAttribute>
        <Name>EventAttribute2</Name>
        <Value>EventAttribute2Value</Value>
      </EventAttribute>
      <!-- Profile -->
    </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: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>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: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>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: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>count</operation>
    </tpm:DXProfileRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
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>
<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. This element can occur multiple times.</td>
</tr>
<tr>
<td>PartnerAttribute</td>
<td>DXPartnerAttribute object that 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 partner.</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 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. This element can occur multiple times.</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 when it creates the account.</td>
</tr>
<tr>
<td>LastUpdatedDate</td>
<td>Date when the account was last updated. Generated by the B2B Data Exchange Server when it updates the account.</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>
</tbody>
</table>
| Status        | Required. DXProfileStatus object that describes the status of the profile. Set this element to one of the following statuses:  
|               | - ENABLED                                                                                         |
|               | - DISABLED                                                                                        |
### WorkflowParameter
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.

### CategoryName
Category of the profile. This element can occur multiple times.

### EventAttribute
Event attribute associated with the profile. The DXEventAttribute object consists of a name and attribute value pair. This element can occur multiple times.

### CreationDate
Date when the partner is created. Generated by the B2B Data Exchange Server when it creates the profile.

### LastUpdatedDate
Date when the partner was last updated. Generated by the B2B Data Exchange Server when it updates the profile.

### 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>
<tr>
<td>Address</td>
<td>Address of the contact.</td>
</tr>
<tr>
<td>PhoneNumber</td>
<td>Telephone number of the contact.</td>
</tr>
<tr>
<td>BusinessPhoneNumber</td>
<td>Business telephone number of the contact.</td>
</tr>
<tr>
<td>FaxNumber</td>
<td>Fax number of the contact.</td>
</tr>
<tr>
<td>EmailAddress</td>
<td>Email address of the contact.</td>
</tr>
</tbody>
</table>

### General B2B Data Exchange Web Service Behavior
The B2B Data Exchange web services cannot modify system objects or properties that identify objects.

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

- The web service operations do not retrieve or modify system objects.
- 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 Data Exchange Server to perform any operation.

You must provide the user credentials for the PowerCenter repository and 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.
- **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 timestamp.

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 user 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
<soapHeader>
  <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>
</soapHeader>
```
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.

B2B Data Exchange Web Services Setup

To set up B2B Data Exchange web services, you must have certain services in the Informatica domain.

When you install B2B Data Exchange, the B2B Data Exchange web services workflows are installed in the `<DX_INSTALLATION_DIR>/powercenter/webservices directory`.

1. Verify that the Informatica domain contains the following services:
   - PowerCenter Repository Service
   - Web Services Hub
   - PowerCenter Integration Service
2. Import the web services workflows into PowerCenter.

Running B2B Data Exchange Web Services on Separate Machine

If the PowerCenter services and the B2B Data Exchange Server run on separate machines, verify that the settings for the B2B Data Exchange Server and Operation Console are set correctly.

In the Informatica Administrator, select the PowerCenter Integration Service that runs B2B Data Exchange workflows. Verify the following environment variable settings:

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DX_SERVER_URL</td>
<td>The RMI URL for B2B Data Exchange Server. For example: <code>rmi://&lt;DXServerHostname&gt;:&lt;RMIPort&gt;/TSSARuntime</code> <strong>Note:</strong> <code>&lt;RMIPort&gt;</code> must match the <code>dx.mii.port</code> parameter in the <code>dx-configuration.properties</code> file. The default port number is 18095.</td>
</tr>
<tr>
<td>DX_CONSOLE_URL</td>
<td>The RMI URL for the B2B Data Exchange Operation Console. For example: <code>rmi://&lt;DXServerHostname&gt;:&lt;RMIPort&gt;/PartnerManagementService</code> <strong>Note:</strong> <code>&lt;RMIPort&gt;</code> must match the <code>dx.tpm.mii.port</code> parameter in the <code>dx-configuration.properties</code> file. The default port number is 18096.</td>
</tr>
</tbody>
</table>
Web Service Import to PowerCenter

You must import the B2B Data Exchange web service workflows into PowerCenter before you can access the web services.

1. Import the following PowerCenter export files:
   ♦ wf_m_DX_WS_TPM.XML. Contains the workflows for the DX_TPM_Partner, DX_TPM_Account, and DX_TPM_Profile web services.
   ♦ wf_m_DX_WS_Endpoint.XML. Contains the workflow for the DX_Endpoint web service.

2. Go to the Web Services Hub console to verify that the B2B Data Exchange web services are correctly imported into PowerCenter.
   If the import process is successful, the list of valid services includes the B2B Data Exchange web services.

You can use the Try-It application in the Web Services Hub console to test the B2B Data Exchange web services. On the XML Input tab, enter the data into the SOAP message and click Send. To avoid authentication errors, do not use the Form Input page to test a B2B Data Exchange web service.

After you verify that the web services are working properly, you can create a client application to send requests to the web services.
APPENDIX A

B2B Data Exchange Java API

This appendix includes the following topics:

- Overview of B2B Data Exchange API, 67
- API Reference, 68

Overview of B2B Data Exchange API

You can use the B2B Data Exchange 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 API methods to process the documents. You can create additional transformations and use the methods in the B2B Data Exchange API to define the transformation behavior. You can also call the API methods from other processes or applications based on your document processing requirements.

The B2B Data Exchange API is contained in the following interface:

```java
com.informatica.b2b.dx.client.DXClient
```

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

For more information about the B2B Data Exchange API methods, see the online API reference:

```
<DXInstallationDir>/powercenter/javadoc\apidocs\com\informatica\b2b\dx\client\DXClient901.html
```

B2B Data Exchange API Versions

B2B Data Exchange 9.0.1 and 9.1 provides the same version of the B2B Data Exchange API. To use the B2B Data Exchange API, include the following line of code in your client application code:

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

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

```java
dxClient.flowEntry(profileId, reuseEventId);
```

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

To use the previous version 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-iface-api=8.6.1-9.1.jar;
  ```
For more information about the methods in the previous version of the B2B Data Exchange 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 B2B Data Exchange API 9.1.

### API Reference

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addEventDocument (String eventId, Document document)</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>completeCorrelation (String eventId, String correlationId)</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>completeCorrelations (String eventId)</td>
<td>void</td>
<td>Closes all correlation events of the specified event Id.</td>
</tr>
<tr>
<td>createChildEvent (String parentEventId, String profileId)</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>createChildEvent (String parentEventId, String profileId, String eventStatusName, String eventTypeName, String subject)</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>flowEntry (String profileId)</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>flowEntry (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>flowExit (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>flowExit (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>Method</td>
<td>Return Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>generateTemporaryFilePath (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>getChildrenCount (String parentEventId)</td>
<td>Integer</td>
<td>Returns the number of child events for a parent event.</td>
</tr>
<tr>
<td>getDocumentData (Document document, Long offset, Long count, String eventId)</td>
<td>byte[]</td>
<td>Returns the data of a document.</td>
</tr>
<tr>
<td>getDocumentFilePath (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>getEventAttribute (String eventId, String attributeKey)</td>
<td>String</td>
<td>Returns the value of an event attribute.</td>
</tr>
<tr>
<td>getEventContext (String eventId)</td>
<td>Map</td>
<td>Returns the run-time context of an event.</td>
</tr>
<tr>
<td>getEventDocuments (String eventId)</td>
<td>List&lt;Document&gt;</td>
<td>Returns information about all documents associated with an event.</td>
</tr>
<tr>
<td>getEventInfo (String eventId)</td>
<td>Properties</td>
<td>Returns the information associated with an event, including event status and event type.</td>
</tr>
<tr>
<td>getEventParentEventId (String eventId)</td>
<td>String</td>
<td>Returns the event ID of the parent event.</td>
</tr>
<tr>
<td>getEventStatusName (String eventId)</td>
<td>String</td>
<td>Returns the status of an event.</td>
</tr>
<tr>
<td>incrementProfileSequenceValue (String profileId, String parameterSequenceId)</td>
<td>String</td>
<td>Increments by one the value of a sequence parameter associated with a profile.</td>
</tr>
<tr>
<td>incrementProfileSequenceValue (String profileId, String parameterSequenceId, String incrementBy)</td>
<td>String</td>
<td>Increments by incrementBy the value of a sequence parameter associated with a profile.</td>
</tr>
<tr>
<td>initiateCorrelation (String eventId, String correlationId, String correlationType, long timeWindowSeconds)</td>
<td>void</td>
<td>Initiates a reconciliation process.</td>
</tr>
<tr>
<td>releaseDelayedEvents (String profileId, List&lt;String&gt; eventIds)</td>
<td>void</td>
<td>Releases delayed events.</td>
</tr>
<tr>
<td>resolveProfileId (String accountNumber, String profileId)</td>
<td>String</td>
<td>Returns the profile ID associated with an application and an account number or partner.</td>
</tr>
<tr>
<td>Method</td>
<td>Return Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>String application, String partnerName, String profileName)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>setEventAttribute ( String eventId, String attributeKey, String attributeName)</td>
<td>String</td>
<td>Sets the value of the attribute associated with an event.</td>
</tr>
<tr>
<td>setEventStatus ( String eventId, String eventStatus)</td>
<td>void</td>
<td>Updates the status of an event.</td>
</tr>
<tr>
<td>setEventStatus ( String eventId, String eventStatusId, String comments)</td>
<td>void</td>
<td>Updates the status of an event.</td>
</tr>
<tr>
<td>setEventSubject ( String eventId, String subject)</td>
<td>void</td>
<td>Updates the subject of an event.</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>
PowerCenter
creating mappings for B2B Data Exchange 5
creating workflows for B2B Data Exchange 6
sources and targets for B2B Data Exchange 2

R
real-time session
configuring Data Exchange workflows 6
reconciled after timeout
reconciliation status 29
reconciliation complete
reconciliation status 29
reconciliation status
ranking 29
reconciled after timeout 29
reconciliation complete 29
reconciliation correlation 29
timed out 29
reconciliations
completing 28
definition 27
initiating 28

S
server plug-in for PowerCenter
installing Data Exchange transformations 2

T
target definition
Data Exchange workflows 4
for B2B Data Exchange 4
timed out
reconciliation status 29

W
workflows
for B2B Data Exchange 6