CHAPTER 3

Data Integration Hub Transformations

This chapter includes the following topics:

- Data Integration Hub Transformations Overview, 27
- DX_Add_Document_To_Event Transformation, 30
- DX_Event_Attribute Transformation, 31
- DX_Event_Details Transformation, 32
- DX_Generate_Temporary_File Transformation, 33
- DX_Notification Transformation, 34
- DX_Publication_Parameters, 36
- DX_Start_Publication Transformation, 36
- DX_Throw_Error, 38

Data Integration Hub Transformations Overview

A Data Integration Hub transformation is a set of functions that process Data Integration Hub data in PowerCenter.

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

Data Integration Hub transformations are custom Java transformations that you use to access the Data Integration Hub API without writing Java code. You can use these transformations, the Unstructured Data transformation, and other transformations to process publications and subscriptions.

The following table describes the Data Integration Hub transformations:

Transformation	Description
DX_Add_Document_To_Event	Attaches a document to an event.
DX_Event_Attrribute	Gets or sets the values of event attributes.

Transformation	Description
DX_Event_Details	Gets or sets the values of a property of an event.
DX_Generate_Temporary_File	Generates a file name for a file in the Data Integration Hub document store.
DX_Notification	Notifies the Data Integration Hub server that processing was successful.
DX_Publication_Parameters	Adds placeholders for the publication instance ID and for the publication date in the Data Integration Hub publication repository.
DX_Start_Publication	Starts a publication process.
DX_Throw_Error	Sets an event status to error if the transformation fails.

Installing and Registering Transformations

To add Data Integration Hub transformations to PowerCenter, use the Data Integration Hub installer.

You install the following components:

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

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

Configuring Transformations

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

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

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

Handling Transformation Errors

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

Each Data Integration Hub transformation uses the following ports to handle errors:

- DXErrorCode. When a transformation fails, the transformation sets the DXErrorCode to a value greater than zero.

- DXErrorMessage. When a transformation fails, the transformation sends an error message that describes the failure to the DXErrorMessage port.

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

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

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

Data Integration Hub Transformations Rules and Guidelines

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

The following list describes rules and guidelines for working with Data Integration Hub transformations:

- Data Integration Hub transformations are based on the PowerCenter Custom transformation and they provide the same configuration options as other custom transformations. You can use them as you use other PowerCenter transformations.
- Data Integration Hub transformations are connected transformations. Connected transformations pass data to or from other transformations.
- Data Integration Hub transformations are passive transformations.
- If a transformation port has a corresponding Data Integration Hub property, the value of the port takes precedence over the value of the property during runtime. When the session runs, if the value of the port is not null, the PowerCenter Integration Service uses the value of the port for processing. If the value of the port is null, the PowerCenter Integration Service uses the value of the Data Integration Hub property for processing.
- Port names are case insensitive and prefix insensitive. DXEventID, dxEVENTid, and eventid are all treated as the same port.
- When you run a PowerCenter workflow that uses a Data Integration Hub transformation, PowerCenter tries to connect to the Data Integration Hub repository to get the list of event statuses and types to use in the transformation. An error message indicates a failed connection. If the connection fails, PowerCenter gets the event type values from the Client plug-in configuration file. To resolve connection errors, verify that the connection section in the dxplugin.ini file contains the following configuration:

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

DX_Add_Document_To_Event Transformation

The DX_Add_Document_To_Event transformation attaches a document to an event.

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

Input Ports

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

The following table describes the DX_Add_Document_To_Event input ports:

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

Input/Output Ports

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

The following table describes the DX_Add_Document_To_Event input/output ports:

Port	Type	Description
DXEventId	string	Required. Identifier of the event to which to attach the document.
DXDataByReference	string	Indicates whether the DXData port contains the document data or a document reference. If the value is true, the DXData port contains a document reference. If the value is null or false, the DXData port contains the document data.
DXDocumentId	string	Identifier of the document to attach to the 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 Integration Hub Properties

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

The following table describes the Data Integration Hub properties of the DX_Add_Document_To_Event transformation:

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

DX_Event_Attribute Transformation

The DX_Event_Attribute gets or sets event attribute values.

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

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

Event names are prefix sensitive. For example, the `DX_event_att1` attribute and the `event_att1` attribute are not treated as the same attribute.

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

Input/Output Ports

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

The following table describes the DX_Event_Attribute input/output ports:

Port	Type	Description
DXEventId	string	Required. Identifier of the event associated with the attribute to get or to update.
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 Integration Hub Properties

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

The following table describes the Data Integration Hub properties of the DX_Event_Attribute transformation:

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

DX_Event_Details Transformation

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

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

Input/Output Ports

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

The following table describes the DX_Event_Details input/output ports:

Port	Type	Description
DXStatusName	string	Status of the event. The status indicates the stages that the event passes during processing.
DXEventId	string	Required. Identifier of the event associated with the properties to get or update.
DXTypeName	string	Type of the event.
DXSubject	string	Subject of the event.
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 Integration Hub Properties

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

The following table describes the Data Integration Hub properties of the DX_Event_Details transformation:

Property	Description
Event Status	Status of the event.
Event Type	Type of the event.
Generate an error in case a failure occurs in this transformation	Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.

DX_Generate_Temporary_File Transformation

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

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

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

Input/Output Ports

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

The following table describes the DX_Generate_Temporary_File input/output ports:

Port	Type	Description
DXEventId	string	Required. Identifier of the event to associate with the generated file.
DXTemporaryFilePath	string	Path and file name of the temporary file.
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 Integration Hub Properties

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

The following table describes the Data Integration Hub properties of the DX_Generate_Temporary_File transformation:

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

DX_Notification Transformation

The DX_Notification transformation sets the status of an event and optionally notifies the Data Integration Hub server that the processing completed.

When you use the transformation in a publication workflow, you can notify the Data Integration Hub server that the publication process is complete and the published data is ready to consume. The Data Integration Hub server then creates child events for the subscriptions and triggers the subscription batch workflows for subscriptions that consume the data when it is published. Scheduled subscription events remain pending until the scheduled subscribers start to consume the data.

The notification that the transformation sends is optional. Therefore, you can use the transformation to change the event status to error without triggering the subscriptions if the publication process ended with errors.

Input/Output Ports

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

The following table describes the DX_Notification input/output ports:

Port	Type	Description
DXEventId	string	Required. Identifier of the event for which you want to set the status. You can get the value from the $$DxeventId parameter.
DXStatusName	string	Name of the status to set. If the value is null, the transformation sets the value based on the transformation properties.
DXNotifySubscribers	string	Required. Indicates whether to send a notification to the Data Integration Hub server. Possible values: - True. The transformation sends a notification to the Data Integration Hub server. The Data Integration Hub server triggers the subscription batch workflow and creates a child event for each subscription. - False. The transformation sets the event status without a notification to the Data Integration Hub server.
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 Integration Hub Properties

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

The following table describes the Data Integration Hub properties of the DX_Notification transformation:

Property	Description
Event Status	Status of the event to create. Default is Complete.
Notify the hub that the publication is ready for subscribers	Indicates whether to send a notification to the Data Integration Hub server that the workflow run ended successfuly. If selected, the Data Integration Hub server creates a child event for each subscription and triggers a subscription batch workflow for subscriptions that you define to consume the data when it is published. Selected by default.
Generate an error in case a failure occurs in this transformation	Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.

DX_Publication_Parameters

The DX_Publication_Parameters transformation adds placeholders for the publication instance ID and for the publication date in the Data Integration Hub publication repository.

You use the transformation to add the placeholders for publications that use custom mappings with real-time workflows.

Input Ports

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

The following table describes the DX_Publication_Parameters input ports:

Port	Type	Description
DXPublicationName	string	Name of the publication. The name must be identical to the name of the publication that the workflow that runs the transformation publishes. The name is not case-sensitive and can contain up to 30 alphanumeric characters and underscores.

Output Ports

Configure output ports for the DX_Publication_Parameters transformation on the **Ports** tab.

The following table describes the DX_Publication_Parameters output ports:

Port	Description
DXPublicationInstanceId	Adds a placeholder for DIH__PUBLICATION_INSTANCE_ID in the Data Integration Hub publication repository. DIH__PUBLICATION_INSTANCE_ID identifies the data set that the application published. When you create a publication real-time workflow, you connect this port to the DIH__PUBLICATION_INSTANCE_ID column of the target.
DXPublicationInstanceDate	Adds a placeholder for DIH__PUBLICATION_INSTANCE_DATE in the Data Integration Hub publication repository. DIH__PUBLICATION_INSTANCE_DATE is the date and time that the application started publishing the data set. When you create a publication real-time workflow, you connect this port to the DIH__PUBLICATION_INSTANCE_DATE column of the target.

DX_Start_Publication Transformation

The DX_Start_Publication transformation instructs the Data Integration Hub PowerCenter Integration Service to trigger the publication workflow for the specified publication.

You use the transformation to start a publication from PowerCenter. For example, if you run a publication pre-process workflow that prepares the data before the publication process can start, add the DX_Start_Publication transformation at the end of the workflow. PowerCenter sends a request to the Data Integration Hub server, and the Data Integration Hub server triggers the publication workflow that publishes the data to the publication repository.

Note: Do not call the DX_Start_Publication transformation more than once in a workflow. If you do, Data Integration Hub starts the publication multiple times.

If you use a PowerCenter parameter file, you can pass the contents of the file to the DXParameterFileContents port and use the parameters in the publication workflow.

Input/Output Ports

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

Note: You can add ports with specific parameters to pass to the workflow. The parameters type must be string. You cannot add Passthrough input ports to the transformation.

The following table describes the DX_Start_Publication input/output ports:

Port	Type	Description
DXPublicationName	string	Optional. Name of the publication to start. The name must match the name of the publication in the Operation Console. **Note:** If you do not define a DXPublicationName port you must define a DXEventId port.
DXEventId	string	Identifier of the event that is associated with the publication pre-process and with the publication. You can get the value from the $$DXEventId parameter. Recommended when a publication pre-process starts a single publication. Ensures that Data Integration Hub uses the same event for both the publication pre-process workflow and the publication workflow and changes the event status accordingly. When a publication pre-process starts multiple publications, do not use the event ID in the DX_Start_Publication transformation. In this case, you can use the Event Details PowerCenter transformation to change the event status. **Note:** If you do not define a DXEventId port you must define a DXPublicationName port.
DXParameterFileContents	string	Parameter names and values from a PowerCenter parameter file. You cannot use a file path in the value of the port.
DXErrorMessage	string	Error message that the transformation generates.
DXErrorCode	string	Error code that the transformation generates. If the transformation fails, the value of the DXErrorCode port is greater than zero.

Data Integration Hub Properties

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

The following table describes the Data Integration Hub properties of the DX_Start_Publication transformation:

Property	Description
Generate an error in case a failure occurs in this transformation	Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.

DX_Throw_Error

The DX_Throw_Error transformation generates an error if an error occurs when the workflow runs.

Use the transformation to perform the following actions:

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

Input Ports

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

The following table describes the DX_Throw_Error input ports:

Port	Type	Description
DXDescription	string	Description of the error to send to the session log. This error is also the description for the log document attached to the event.
DXMessageType	string	Optional. Type of the error event. Alphanumeric value to associate with the event. Any value is valid.
DXMIMEType	string	MIME type of the document to attach to the event.

Input/Output Ports

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

The following table describes the DX_Throw_Error input/output ports:

Port	Type	Description
DXEventId	string	Required. Identifier of the event that is associated with the error.
DXData	string binary text	Log document to attach to the event. This port can contain the data of the document or a file path to the document. If the value of the parameter is null, the transformation creates an empty document and adds the document to the event. To attach a document with text data, set the datatype of the port to string or text. To attach a document with binary data, change the datatype of the port to binary.
DXDataByReference	string	Indicates whether the DXData port contains the document data or a document reference. Possible values: - True. The DXData port contains a document reference. - False. The DXData port contains the document data. - Null. The DXData port contains the document data.

Port	Type	Description
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 Integration Hub Properties

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

The following table describes the Data Integration Hub properties of the DX_Throw_Error transformation:

Property	Description
Error log document description	Description of the error log document that the transformation attaches to the event.
Message type	Alphanumeric value to associate with the event. You can enter any value.
Generate an error in case a failure occurs in this transformation	Indicates whether to set the status of the event to error when the transformation generates an error. Selected by default.

CHAPTER 4

Workflows

This chapter includes the following topics:

- Workflows Overview, 40
- Batch Workflows, 41
- Real-time Workflows, 41
- Developing Publication Real-time Workflows, 42
- Managing Workflows in the Operation Console, 45

Workflows Overview

A workflow is a set of instructions that define how to process data that applications publish and consume from the Data Integration Hub publication repository. A Data Integration Hub workflow represents a PowerCenter workflow and mapping.

Before you create a workflow in Data Integration Hub, you must save the PowerCenter workflow to the PowerCenter repository. Optionally, you can export the workflow from PowerCenter to an XML file that will be used as the workflow definition file. You must use the Repository Manager to export the workflow.

Note: Do not save the workflow to a folder where the folder name is prefixed by `DIH__pub` or by `DIH__sub`.

When you create a publication or a subscription with a custom mapping, you select a Data Integration Hub workflow to use for the custom mapping. You do not need to create a workflow for a publication or a subscription that uses an automatic mapping. Data Integration Hub creates the PowerCenter workflow and mapping directly from the publication.

To use the workflow in Data Integration Hub as a publication workflow or a subscription workflow, create a Data Integration Hub workflow in the Data Integration Hub Operation Console by selecting the workflow in the PowerCenter repository or by selecting the exported workflow definition file. For more information, see "Managing Workflows in the Operation Console" on page 45.

You can use PowerCenter workflows to define topic structures by extracting table structures from the workflow.

Batch Workflows

Batch workflows run according to a trigger and not continuously. Data Integration Hub uses batch workflows to process publications and subscriptions. Batch workflows use the PowerCenter Web Services Hub to communicate with PowerCenter.

The following operations can trigger a batch workflow:

- Scheduled publication or subscription starts.
- Operator manually runs the workflow.
- DX_Start_Publication transformation sends a request to the Data Integration Hub server.
- DX_Notification transformation sends a request to the Data Integration Hub server to start subscription workflows for a completed publication.

When an operation triggers a batch workflow, Data Integration Hub creates an event with a processing status. If the trigger occurs for a scheduled subscription, the event for the subscription has a delayed status from the time the data is ready to consume until the scheduled subscription start time.

Real-time Workflows

Real-time workflows process data from real-time sources on-demand. They read data from real-time sources and write the data to Data Integration Hub. For example, data that is published through web-service providers or Java Message Service (JMS) queues. Data Integration Hub uses real-time workflows to process publications.

Real-time workflows are not started by Data Integration Hub. Running and maintaining the workflows is the responsibility of the Data Integration Hub developer. You can run real-time workflows continuously or on-demand.

When publishing through a real-time workflow, you group the published data into single publications at predefined time intervals. At the scheduled publication time, Data Integration Hub creates an event with a *processing* status.

Real-time Workflows Rules and Guidelines

When you develop PowerCenter real-time workflows to use in publications with custom mappings, consider the following rules and guidelines:

- Make sure to name and store PowerCenter entities for custom mappings in different folders with a different naming convention from the naming convention of PowerCenter entities for automatic mappings.
- You must enable the property **Enable high precision** in the session that writes data to the Data Integration Hub publication repository.
- To prevent naming conflicts, do not use **_DIH__** in the parameter names, and do not use workflow and mapping parameters with the same names as workflow and mapping parameters that Data Integration Hub uses in workflows for publications with automatic mappings.
- The workflows do not generate error messages. Maintaining the workflows is the responsibility of the Data Integration Hub developer.

- If you have multiple tables in a topic and you want to ensure that the records for all tables are included in the same publication, change the value of the system property **dih.realtime.time.window**. For information, see the *Data Integration Hub Administrator Guide*.

Developing Publication Real-time Workflows

You develop PowerCenter workflows for Data Integration Hub in the same way that you develop other PowerCenter workflows. Data Integration Hub transformations interact directly with Data Integration Hub.

Before you develop Data Integration Hub workflows in PowerCenter, verify that the Data Integration Hub PowerCenter Client and server plug-ins are installed and registered to the PowerCenter repository.

The Data Integration Hub operator then creates a publication in the Data Integration Hub Operation Console, and selects the Data Integration Hub publication workflow. For more information, see the *Data Integration Hub Operator Guide*.

Developing Publication Real-time Workflows Process

To develop a workflow in PowerCenter that processes Data Integration Hub real time publications, perform the following tasks:

1. Create the source and target definitions.
2. Create the mapping and add transformations.
3. Create the workflow and the session and save the workflow.
4. In the Data Integration Hub Operation Console, create a topic to which the application will publish the real-time data. Base the topic structure on the PowerCenter workflow or on the workflow definition file. Data Integration Hub creates the topic in PowerCenter and names the target according to the Data Integration Hub naming conventions.
5. In the workflow that you created in step 3, rename the target to the name of the topic target in PowerCenter and save the workflow. Optionally, export the workflow to an XML workflow definition file.
6. In the Data Integration Hub Operation Console, create a publication real-time workflow and select the PowerCenter workflow that you modified in step 5.

Step 1. Create the Source and Target Definitions

When you develop a publication real-time workflow, you create the source based on the structure of the JMS or web service source that you want to publish and set the target to the Data Integration Hub publication repository. You can also add properties to store other Data Integration Hub values that you want to send back from PowerCenter.

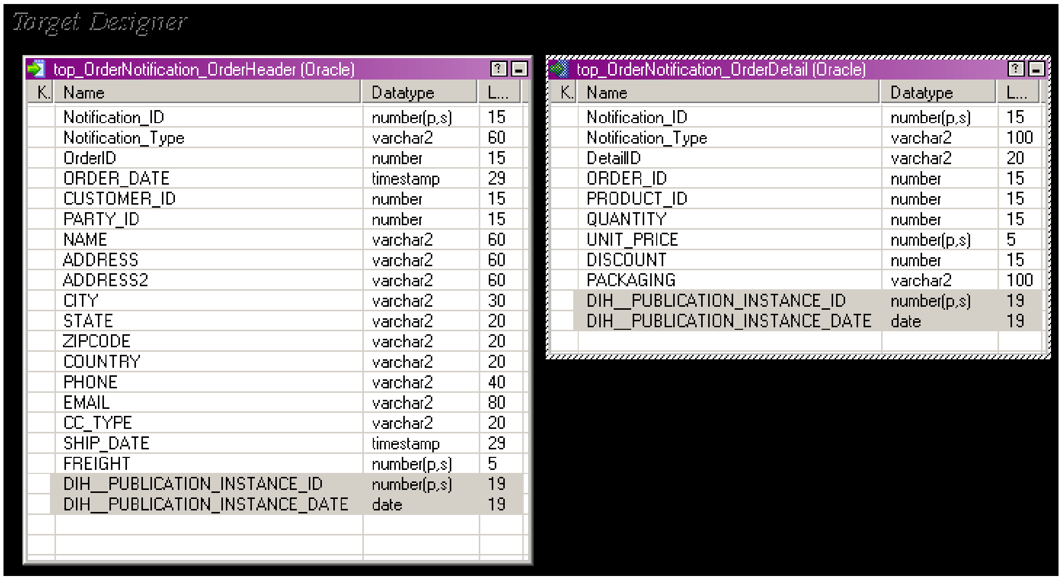
You create the source and the target definitions in the PowerCenter Designer in the same way that you create source and target definitions for other mappings. For general information about source and target definitions, see the *PowerCenter Designer Guide*.

When you create the source and the target definitions, consider the following guidelines:

- Use the DIH__STAGING connection in the target definition.
- The database target type must match the database type of the Data Integration Hub publication repository.

- The target must contain DIH__PUBLICATION_INSTANCE_ID and DIH__PUBLICATION_INSTANCE_DATE columns.

The following image shows an example of a target with the DIH__PUBLICATION_INSTANCE_ID and DIH__PUBLICATION_INSTANCE_DATE columns:



Step 2. Create the Mapping

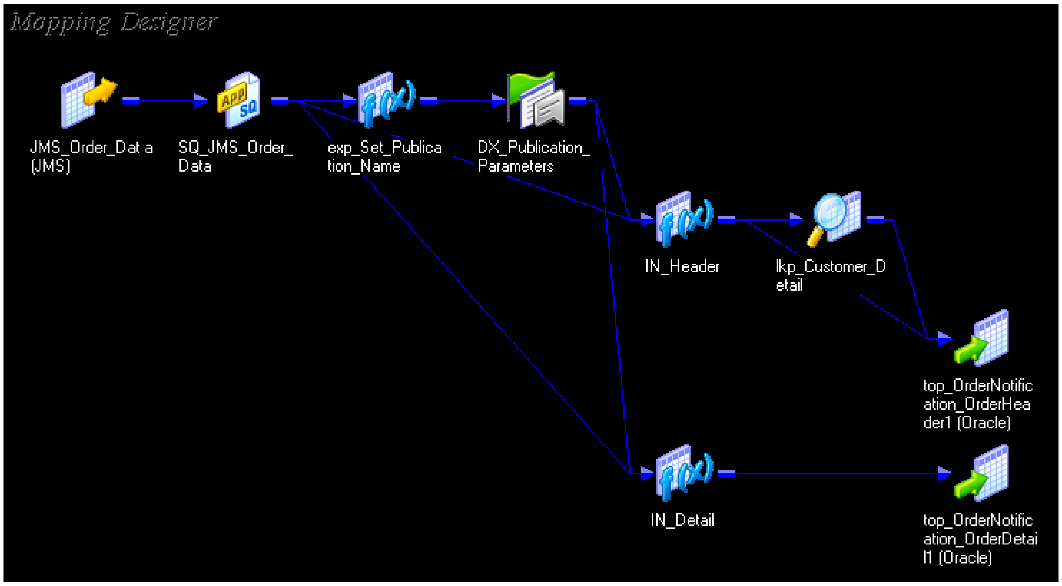
Create a mapping that contains the source definition, the target definition, and the transformations that you want to use in the real-time publication workflow.

You create a mapping for Data Integration Hub in the same way that you build other PowerCenter mappings.

When you create the source and the mapping, consider the following guidelines:

- Add the DX_Publication_Parameters transformation to the mapping. You can also add other PowerCenter transformation to the mapping.
- In the DXPublicationName port of the DX_Publication_Parameters transformation enter the name of the publication to which the workflow publishes. The DX_Publication_Parameters transformation adds placeholders for the publication instance ID and for the publication date in the Data Integration Hub publication repository.
- Connect the port DXPublicationInstanceId from the DX_Publication_Parameters transformation to the DIH__PUBLICATION_INSTANCE_ID column in the target.
- Connect the port DXPublicationInstanceDate from the DX_Publication_Parameters transformation to the DIH__PUBLICATION_INSTANCE_DATE column in the target.

The following image shows an example mapping that includes the DX_Publication_Parameters transformation. The mapping also includes an expression that sets the publication name:



Step 3. Create and Save the PowerCenter Workflow and Session

Create the real-time workflow and the session that runs the mapping. You create the workflow in the same way that you create other PowerCenter workflows.

1. In the Workflow Designer, create the workflow and the session object. You must configure the session as a real-time session. For general information about creating workflows and sessions, see the *PowerCenter Advanced Workflow Guide*.
2. Save the session object.
3. Test the workflow to ensure that it works correctly.
4. Save the workflow to the PowerCenter repository. Optionally, use the Repository Manager to export the workflow from PowerCenter to an XML file.

   Note: Do not save the workflow to a folder where the folder name is prefixed by `DIH__pub` or by `DIH__sub`.

Step 4. Create the Topic

Create the topic to which the application will publish the real-time data. The topic defines the data structure and additional data definitions such as the data retention period.

You create the topic in the Data Integration Hub Operation Console.

When you create the topic, consider the following guidelines:

- You define the topic structure in the **Structure** page of the **Topic** wizard.
- From the **Add Tables** list, choose **From PowerCenter**.
- In the **Add Tables from PowerCenter Workflow** dialog box, choose one of the following options:
  - Choose **Select a workflow from the PowerCenter repository** to add tables from the real-time workflow.

- Choose **Select a workflow definition file (.xml)** to add tables from the XML file that you exported from the real-time workflow.

For more information, see the *Data Integration Hub Operator Guide*.

Step 5. Rename the Workflow Target

Rename the target of the PowerCenter real-time workflow to the name of the topic target in PowerCenter. You rename the target in the PowerCenter Designer.

1. In the navigation pane, access the folder of the topic that you created and expand the folder tree.
2. Expand the `Targets` subfolder and copy the target name into the clipboard.
3. Access the folder of the workflow and expand the folder tree.
4. Expand the `Targets` subfolder and replace the target name with the name that you copied from the topic folder.
5. Save the session object.
6. Test the workflow to ensure that it works correctly.
7. Save the PowerCenter workflow to the PowerCenter repository. Optionally, use the Repository Manager to export the workflow from PowerCenter to an XML file.

   Note: Do not save the workflow to a folder where the folder name is prefixed by DIH__pub or by DIH__sub.

Step 6. Create the Publication Real-time Workflow

Create the Data Integration Hub real-time workflow that will publish the real-time data. The Data Integration Hub operator can use the workflow to create a real-time publication.

You create the workflow in the Data Integration Hub Operation Console, in the **Workflows** page.

When you create the workflow, consider the following guidelines:

- You define the topic structure in the **Structure** page of the Topic wizard.
- In **Flow Type**, choose **PowerCenter Real-time workflow**.
- In **Type**, choose **Publication**.

The Data Integration Hub operator selects the workflow in the **Create Custom Real Time Publication** wizard, in the **Mapping** page.

Managing Workflows in the Operation Console

Use the Navigator to add, edit, and delete Data Integration Hub workflows that you use for custom mappings. Each workflow represents a PowerCenter workflow.

1. In the Navigator, click **Hub Management** > **Workflows**.

   The **Workflows** page appears. You can sort the display by either the **Workflow Name** or the **Last Modified** column.
2. Choose to create, edit, or delete a workflow.

- To create a workflow, click **New Workflow** and define the workflow properties. For more information, see "Workflow Properties" on page 46.
- To edit a workflow, click the **Edit** icon next to the workflow that you want to edit and change the workflow properties.
- To delete a workflow, click the **Delete** icon next to the workflow that you want to delete and confirm the deletion.

  Note: If you delete the workflow, you also delete all of the dependent objects for that workflow.

3. If the workflow contains parameters, you can add values to the parameters on the **Workflow Parameters** tab of the **Create Workflow** window. You can customize the layout and behavior of the parameters with the Forms Designer. For more information, see "Customizing Workflow Parameters" in the chapter "Forms Designer".

   Note: If the workflow contains session parameters, and you do not want the Data Integration Hub operator to set the session parameters in the Publication wizard or the Subscription wizard, use the Forms Designer to hide the session parameters.
4. To associate event attributes with the workflow, add event attributes on the **Event Attributes** tab.
5. To associate a workflow with a publication or a subscription, use the Subscription or Publication wizard and select **Custom Mapping**.

Workflow Properties

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

A workflow definition contains the following properties:

Workflow Name

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

Description

Description of the workflow.

Flow Type

Select the type of workflow to process Data Integration Hub publications and subscriptions with PowerCenter.

- Select **PowerCenter Batch workflow** to associate the workflow with a PowerCenter batch workflow. A PowerCenter batch workflow reads from a file, database, or another source. The workflow runs once and stops after completion.
- Select **PowerCenter Real-time workflow** to associate the workflow with a real-time workflow. A PowerCenter real-time workflow runs continuously and reads from real-time sources. Use a real-time workflow to read data from real-time sources and write the data continuously to Data Integration Hub. For example, use it for data that is published through web-service providers or Java Message Service (JMS) queues. When publishing through a real-time workflow, you can group the published data into single publications at predefined time intervals.

Type

Type of process for which the workflow is used. Select one of the following options:

- Publication Pre-Processing. Runs a publication pre-process.
- Publication. Runs a publication process.
- Subscription. Runs a subscription process.

PowerCenter Workflow

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

- **Select a workflow from the PowerCenter repository**. Select a PowerCenter workflow that you saved in the PowerCenter repository. To show all workflows in the repository, click **Show All**. To search for a workflow by name, enter a string in the **Folder Name** text box and then click **Search**.
- **Select a workflow definition file (XML)**. Select an XML PowerCenter workflow definition file that you exported from the PowerCenter repository. To select a file, click the browse button, browse to the location of the required file, select the file, and then click **Upload**.

CHAPTER 5

Forms Designer

This chapter includes the following topics:

- Forms Designer Overview, 48
- Forms Designer User Interface, 48
- Element Properties, 50
- Group Properties, 51
- Customizing Workflow Parameters, 51

Forms Designer Overview

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

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

Forms Designer User Interface

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

The **Designer** tab displays information about the elements, such as label and type. The information appears in read-only mode.

The following table describes the element properties that you can view on the **Designer** tab:

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

Forms Designer Actions

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

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

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

Element Properties

When you edit an element in the **Details** dialog box, you can change element properties such as label, type, and default value.

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

The following table describes the element properties:

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

Group Properties

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

The following table describes the group properties:

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

Customizing Workflow Parameters

Use the Forms Designer to customize user interface layout and field behavior for workflow parameters in the Operation Console.

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

1. In the Navigator, click **Hub Management** > **Workflows**.
2. Choose to create or edit a workflow.
   - To create a workflow, import that workflow XML file from PowerCenter. Ensure that the workflow you import contains workflow, mapping, or session parameters.
   - To edit a workflow, click **Edit** next to the workflow that you want to edit.
3. Click the **Workflow Parameters** tab.

   The **Preview** pane displays the default view of the workflow parameter list.
4. Click **Designer**.

   The **Designer** pane displays the parameter list in alphabetical order.
5. To arrange the parameters and groups, drag the element or group you want to arrange.
6. To change the behavior and appearance of a parameter element, click the parameter and click **Actions** > **Edit**.

   The **Details** dialog box appears.
7. Change the properties for the parameter and click **Save**.

   Note: The **Label** property is a required property.
8. To add a group, click the **Actions** menu and choose to add a tab, column, header, or disclosure.
9. To preview the customized form, click **Preview** and verify the appearance of the form and any field dependencies you defined.

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

    The customized form appears when you create or edit a publication or subscription and choose the workflow for a custom mapping.

CHAPTER 6

Data Integration Hub Run Publication Subscription APIs

This chapter includes the following topics:

- Data Integration Hub Run Publication Subscription APIs Overview, 53
- Data Integration Hub Event Status API, 54
- Data Integration Hub Run Publication Subscription Web Service API, 56
- Data Integration Hub Run Publication Subscription Command Line API , 58

Data Integration Hub Run Publication Subscription APIs Overview

Use the Data Integration Hub Run Publication Subscription APIs to start the running of a specific publication and of a specific subscription by an external trigger.

You can use the following Data Integration Hub APIs to start the running of a publication or of a subscription:

- Web service API. Starts a publication or a subscription from any web service client application. You can run the web service API from the Data Integration Hub server or from a remote server.
- Command line API. Starts a publication or a subscription from a command line utility. You can run the command line API from the Data Integration Hub server.

The Run Publication Subscription APIs return the status of the action that you perform. If the running of a publication or of a subscription succeeds, the APIs return the event ID of the publication or the subscription event that Data Integration Hub generates. You can run the Data Integration Hub Event Status API to query the status of the publication or subscription event.

To start the running of a publication or of a subscription from an API, the following conditions must be true:

- The Data Integration Hub operator must configure the schedule of the publication or of the subscription with the option **Manually or by an external trigger**. For more information, see the *Data Integration Hub Operator Guide*.
- You must use a Data Integration Hub user with permissions and privileges for the publication or for the subscription that you want to run. For more information, see the *Data Integration Hub Administrator Guide*.

Data Integration Hub Event Status API

When you use a Data Integration Hub Run Publication Subscription API to start the running of a publication or of a subscription and the action succeeds, Data Integration Hub returns the event ID of the publication or the subscription event that it generates.

The manner in which Data Integration Hub returns the event ID depends on the API that you use to run the publication or the subscription:

- When you run the web service API, Data Integration Hub returns the event ID in the web service action status.
- When you run the command line API, Data Integration Hub returns the event ID in the command line notification.

You can use the Data Integration Hub Event Status API to query the status of the publication or subscription event according to the event ID. You can see whether the publication or subscription process is still running, and after the process is complete, you can see whether it completed successfully. If the process fails, the response to the query includes the cause of the failure.

To query the status of an event, use the following REST URL:

```
http://<DIH_console_host_port>/dih-console/api/v1/event/eventId
```

For example:

```
http://localhost:18080/dih-console/api/v1/event/2435
```

Sample Files

The `<DIH_InstallationDir>\samples\api\restapi\examples\event` folder contains the following files:

- `EventResponse.java`: Event response model class.
- `HttpRestClientEventApi.java`: A simple HTTP client that is based on Java API to query event status. The client uses `org.codehaus.jackson.map.ObjectMapper` to parse event status and send a `json` response to an `EventResponse` object. For more information, see the following websites:
  - http://wiki.fasterxml.com/JacksonDownload
  - http://mvnrepository.com/artifact/org.codehaus.jackson
- `SprintRestTemplateClientEventApi.java`: A sample client that uses a SpringRest template to query event status. Use this client if your client applications are contained in a Spring container.

Data Integration Hub Event Status API Response

When you use the Data Integration Hub Event Status API to query the status of a publication or a subscription event, the API returns the event response in an `EventResponse.java` model class.

The response includes the following information:

Property	Description
responseCode	Response of the Run Publication Subscription API action.
eventId	ID of the event that Data Integration Hub generates for the publication or for the subscription.
eventType	Type of the event that Data Integration Hub generates for the publication or for the subscription.

Property	Description
topicName	Name of the topic that is associated with the publication or with the subscription.
publicationName or subscriptionName	Name of the publication or of the subscription.
applicationName	Name of the publishing or of the subscribing application.
eventStatus	Status of the event that Data Integration Hub generates for the publication or for the subscription.
eventStartTimeLong	Time when the publication or the subscription event started. System time in milliseconds as returned by Java API `java.lang.System.currentTimeMillis`.
eventEndTimeLong	Time when the publication or the subscription event ended. System time in milliseconds as returned by Java API `java.lang.System.currentTimeMillis`.
referencedEventsList	Applicable for file publication events, aggregated subscription events, and compound subscription events. List of event IDs that are related to the file publication, the aggregated subscription, or the compound subscription event. For example, the referencedEventsList of a file publication event includes the file events of the files that are published as part of the publication event.
isFinal	Is the event in a final state.
isError	Is the event in Error status.
detailedMessage	Applicable for events in an Error status. If the error is caused by Data Integration Hub, `detailedMessage` returns the error message from the Data Integration Hub event. For any other error, for example an authentication failure or an incorrect REST URL request, `detailedMessage` includes a message that describes the cause of the error.

Sample Event Status API Responses

Response to a request to query the status of publication event 4003:

```
{
    "responseCode": "SUCCESS",
    "eventId": 4003,
    "eventType": "Publication",
    "topicName": "top_120",
    "publicationName": "ng_pub_120_1",
    "applicationName": "app1",
    "eventStatus": "Complete",
    "eventStartTimeLong": 1431078308560,
    "eventEndTimeLong": 1431078313780,
    "isFinal": true,
    "isError": false
}
```

Response to a request to query the status of aggregated subscription event 3009, which includes subscription events 3008 and 3007:

```
{
    "responseCode": "SUCCESS",
    "eventId": 3009,
    "eventType": "Aggregated Subscription",
```

```
    "topicName": "topic1",
    "subscriptionName": "sub1",
    "applicationName": "app1",
    "eventStatus": "Complete",
    "eventStartTimeLong": 1431065700088,
    "eventEndTimeLong": 1431065704372,
    "referencedEventsList": "3008,3007"
    "isFinal": true,
    "isError": false
}
```


Response to a request to query the status of publication event 3016, where the publication process failed:


```
Response:

{
    "responseCode": "SUCCESS",
    "eventId": 3016,
    "eventType": "Publication",
    "topicName": "top_120",
    "publicationName": "ng_pub_120_1",
    "applicationName": "app1",
    "eventStatus": "Error",
    "eventStartTimeLong": 1431066353202,
    "eventEndTimeLong": 1431066357162,
    "isFinal": true,
    "isError": true,
    "detailedMessage": "Error while copying several rows :\nSrcFailedRows:
1\nTgtFailedRows: 1\nSrcSuccessRows: 2\nTgtSuccessRows: 2\nPowerCenter workflow:
s__DIH_pub_ng_pub_120_1\nPowerCenter session: s__DIH_pub_ng_pub_120_1\n\nCheck the
PowerCenter session log for more information."
}
```


Data Integration Hub Run Publication Subscription Web Service API

Use the Data Integration Hub Run Publication Subscription web service API to run a specific publication or a specific subscription from a web service client application. You can call the web service operations from the Data Integration Hub server or from a remote server. The API is available on Windows and UNIX operating systems.

Description of the Run Publication Subscription Web Service API

When you use the Run Publication Subscription web service API, consider the following information:

SOAP Request

The SOAP request contains the following properties:

Property	Description	Values
UsernameToken	Name and password of a user with privileges on the PowerCenter Web Services Hub.	Name and password.
name	Required. Name of the publication or subscription to trigger.	Name. **Note:** Names are case sensitive.
type	Required. Type of entity to trigger.	`publication` `subscription`

preprocessPublication	Optional. Determines whether a publication pre-process runs before the publication process runs.	`true`
		`false`
		Note: If you enter a different value, a null pointer exception might appear in the response.

parameterList	Optional. Additional parameters.	List of parameter values as key-value pairs.
paramFileContent	Optional. Parameters and variables that can you can reference in sessions and in workflows.	Parameters and variables as strings.

SOAP Response

When complete, the SOAP response displays the following lines:

```
<n4:Envelope xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:n="http://
b2b.informatica.com/dih/962/services" xmlns:n4="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:n5="http://schemas.xmlsoap.org/wsdl/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
   <n4:Body>
      <n:StartPubSubResult>
         <status>SUCCESS</status>
      </n:StartPubSubResult>
   </n4:Body>
</n4:Envelope>
```

Rub Publication Subscription Web Service Action Status

When you use the Data Integration Hub web service API to start the running of a publication or of a subscription, Data Integration Hub returns the status of the action that you perform.

Running a publication or a subscription from the web service API returns one of the following statuses:

- Success. Data Integration Hub triggered the publication or the subscription successfully. The status message includes the event ID of the publication or the subscription event that Data Integration Hub generates.
- Failure. Data Integration Hub could not trigger the publication or the subscription.

- Publication in process. Applicable to relational database publications only. Data Integration Hub cannot run the publication because the publication is currently running.
- No publications to consume. Applicable to subscriptions only. Data Integration Hub did not run the subscription because no publications are ready for consumption by the subscription.

Run Publication Subscription Web Service Security

When you send a request to the Run Publication Subscription web service, you must provide the user credentials to access the PowerCenter repository.

User Authentication

A client application that calls a Data Integration Hub web service must log in to the PowerCenter repository to perform any operation.

You must provide the user credentials for the PowerCenter repository in the SOAP message header of the web service request.

The message header contains a *Security* element. The *Security* element contains the user name and password to log in to the PowerCenter repository that stores the Data Integration Hub web service. The user account must have permissions to run the Data Integration Hub web service. You can use plain text passwords and digested passwords. For more information about configuring digested passwords in PowerCenter, see the *Informatica PowerCenter Web Services Provider Guide*.

Data Integration Hub Run Publication Subscription Command Line API

Use the Data Integration Hub Run Publication Subscription command line API to start the running of a specific publication and of a specific subscription from a command line utility. The API is available on Windows and UNIX operating systems.

You can run the Run Publication Subscription command line API from the Data Integration Hub server.

Data Integration Hub Run Publication Subscription Command Line API Command Syntax

The Data Integration Hub Run Publication Subscription command line API uses the following syntax:

```
runsubscriptionpublication
<-c|--command> publication|subscription
<-n|--name> name
<-u|--user> user
<-p|--password> password
[-r|--preprocess]
[-f|--file "<parameters file path>"]
[-v|--parameter "<key>=<value>"...]
[--server "<hostname:port>"]
```

Note: When you run a publication or a subscription with both the `--file` and the `--parameter` options, Data Integration Hub orders the parameters according to the following sequence:

1. Parameters defined in the workflow-specific section of the parameters file that you specify in `--file`.

2. Parameters that you define in `--parameter`.
3. Parameters defined in the Global section of the parameters file that you specify in `--file`.

The command line API is in the following location: `<DIHInstallationDir>/dx-tools`

The following table describes the Data Integration Hub Run Publication Subscription command line API options and arguments:

Option	Argument	Description
-c --command	command	Required. Command to run the publication or to run the subscription. Enter one of following commands: - publication. Instructs Data Integration Hub to trigger the publication workflow for the publication that you define in the `--name` option. - subscription. Instructs Data Integration Hub to trigger the subscription workflow for the subscription that you define in the `--name` option.
-n --name	name	Required. Name of the publication to run or of the subscription to run. Data Integration Hub verifies the permissions and privileges to run the publication or the subscription against this user.
-u --user	user name	Optional. User name of an Operation Console user account with the Manage Data privileges to run the `runsubscriptionpublication` command. To run a subscription or a publication with the command, the user account must have run subscription or run publication privileges. If you use Informatica domain authentication or Informatica domain with Kerberos authentication, the user name must specify the Informatica security domain, separated by the @ symbol. For example: `Administrator@SecurityDomain`
-U	Environment variable	Optional. Environment variable that contains a user name. User name of an Operation Console user account with the Manage Data privileges to run the `runsubscriptionpublication` command. To run a subscription or a publication with the command, the user account must have run subscription or run publication privileges. If you use Informatica domain authentication or Informatica domain with Kerberos authentication, the user name must specify the Informatica security domain, separated by the @ symbol. For example: `Administrator@SecurityDomain` **Note:** You must specify at least one of the user name options, -u or -U.
-p --password	password	Optional. Password for the Operation Console user that runs the `runsubscriptionpublication` command.
-P	Environment variable	Optional. Environment variable that contains the a password. Password for the Operation Console user that runs the `runsubscriptionpublication` command. The password must be encrypted. Use `dxencrypt` for the encrypted value. **Note:** You must specify at least one of the password options, -p or -P.
-r --preprocess	-	Optional for the publication command. For publications where a pre-process is defined, this command runs the pre-process.

Option	Argument	Description
-f --file	parameters_file_path	Optional. A path to a workflow parameter file. Data Integration Hub uses the content of the file as the content of the workflow parameter file.
-v --parameter	<key>=<value>	Optional. A key-value pair pass as a custom workflow parameter: - When the API runs a publication, Data Integration Hub passes the parameter to the publication workflow. - When the API runs a publication pre-process, using the `--preprocess` option, Data Integration Hub passes the parameter to the pre-process workflow. - When the API runs a subscription, Data Integration Hub passes the parameter to the subscription workflow. You can enter multiple parameters. For example: `-c subscription -v "employee_ID=148" -v "department=sales"` **Note:** The parameter or parameters must be defined for the workflow of the entity that the API runs. Data Integration Hub does not pass undefined parameters to PowerCenter, and does not display an error.
--server	hostname:port	Optional. Host name and port number of the Data Integration Hub server. If you do not enter a value, the Run Publication Subscription API connects to the localhost server with the default port 18095. You must enclose the value in quotation marks. For example: `-c subscription --server "localhost:18096"`

Run Publication Subscription Command Line API Notifications

When you use the Data Integration Hub command line API to start the running of a publication or of a subscription, Data Integration Hub notifies you whether the command has succeeded or not.

If the command succeeds, the notification includes the event ID of the publication or the subscription event that Data Integration Hub generates.

If the command fails, the notification includes an error message with the cause of the failure. A failure can be caused by one of the following reasons:

- The publication or the subscription is disabled.
- Publication in process. Applicable to relational database publications only. Data Integration Hub cannot run the publication because the publication is currently running.
- No publications to consume. Applicable to subscriptions only. Data Integration Hub did not run the subscription because no publications are ready for consumption by the subscription.

Chapter 7

Data Extraction APIs

This chapter includes the following topics:

- Data Extraction APIs Overview, 61
- Data Integration Hub Catalog API, 61
- Data Integration Hub Events View, 64

Data Extraction APIs Overview

Use the data extraction APIs to extract data from Data Integration Hub.

You can use the following Data Integration Hub APIs for data extraction:

- Catalog API. Extracts data from the Data Integration Hub catalog, including topic, publication, and subscription metadata.
- Events API. Provides a database view of Data Integration Hub events.

Data Integration Hub Catalog API

Use the Catalog API to extract data from the Data Integration Hub catalog, including topic metadata and metadata about the publications and subscriptions that are associated with each topic.

You can extract metadata pertaining to topics, publications, and subscriptions for which you have both View privileges and Read access permissions.

To extract data from the catalog, use the following REST URL:

```
http://localhost:18080/dx-console/api/v1/catalog/topics
```

Data Integration Hub Catalog API Response

When you use the Data Integration Hub Catalog API to extract data from the Data Integration Hub catalog, the API returns a JASON string that contains metadata about all the topics for which you have the required privileges and permissions.

The string includes the following data for each topic in the response:

topicName

Name of the topic.

topicDesc

Textual description of the topic.

topicType

Type of the topic: Delta or Full.

topicTables

For each table in th topic, an entry with the table name.

publications

For each publication that is associated with the topic, the following data is provided:

publicationName

Name of the publication.

publicationDesc

Textual description of the publication.

applicationName

Application from which the publication publishes data or files.

publicationSourceType

Type of publication source.

publicationConnectionName

For relational database publications and for HDFS publications: name of the connection from where the publication workflow reads the data or the files to be published.

publicationDBType

For relational database publications: type of database.

subscriptions

For each subscription that is associated with the topic, the following data is provided:

subscriptionName

Name of the subscription.

subscriptionDesc

Textual description of the subscription.

applicationName

Application that consumes data or files from the topic.

subscriptionTargetType

Type of subscription target.

subscriptionConnectionName

For relational database subscriptions and for HDFS subscriptions: name of the connection to where the subscription workflow writes the data or the files that the application consumes.

subscriptionDBType

For relational database subscriptions: type of database.

Sample Data Integration Hub Catalog API Response

The following example shows a response to a request to extract data from the Data Integration Hub catalog:


```
{
    "responseCode": "SUCCESS",
    "catalogTopics": [
        {
            "topicName": "FileTopic",
            "topicDesc": null,
            "topicType": "Delta",
            "topicTables": [
                {
                    "tableName": "Orders"
                }
            ],
            "publications": [
                {
                    "publicationName": "FilePub",
                    "publicationDesc": null,
                    "applicationName": "FileApp",
                    "publicationSourceType": "FLAT_FILE",
                    "publicationConnectionName": null,
                    "publicationDBType": null
                }
            ],
            "subscriptions": [
                {
                    "subscriptionName": "FileSub",
                    "subscriptionDesc": null,
                    "applicationName": "FileApp",
                    "subscriptionTargetType": "FLAT_FILE",
                    "subscriptionConnectionName": null,
                    "subscriptionDBType": null
                }
            ]
        },
        {
            "topicName": "OrderTopic",
            "topicDesc": null,
            "topicType": "Delta",
            "topicTables": [
                {
                    "tableName": "OrderTable"
                },
                {
                    "tableName": "CustomerTable"
                },
            ],
            "publications": [
                {
                    "publicationName": "OrdersPublication",
                    "publicationDesc": null,
                    "applicationName": "OrderPublications",
                    "publicationSourceType": "RDBMS",
                    "publicationConnectionName": "OrderSource",
                    "publicationDBType": "SQLSERVER"
                }
            ],
            "subscriptions": [
                {
                    "subscriptionName": "OrdersSubscription",
                    "subscriptionDesc": null,
                    "applicationName": "OrderSubscriptions",
                    "subscriptionTargetType": "FLAT_FILE",
                    "subscriptionConnectionName": null,
                    "subscriptionDBType": null
                },
                {
                    "subscriptionName": "OrderSubs",
                    "subscriptionDesc": null,
```

```
                        "applicationName": "OrderSubscriptions",
                        "subscriptionTargetType": "RDBMS",
                        "subscriptionConnectionName": "OrderTarget",
                        "subscriptionDBType": "SQLSERVER"
                    }
                ]
            }
                ]
            }
        ]
    }
```

Data Integration Hub Events View

Use your SQL client to view Data Integration Hub events in the Data Integration Hub repository.

In your SQL client, the events view resides under the Data Integration Hub database, in **DX_VIEW_PUB_SUB_EVENT**.

The view contains the following columns:

EVENT_ID

ID of the event.

EVENT_TYPE

Type of event, such as publication, subscription, file, and system events.

PUBLICATION_SUBSCRIPTION_NAME

Name of the publication or of the subscription that generated the event.

APPLICATION_NAME

Name of the application that is associated with the publication or of the subscription that generated the event.

TOPIC_NAME

Name of the topic that is associated with the publication or of the subscription that generated the event.

EVENT_STATUS

Status of the event, such as error, complete, and pending events.

EVENT_START_TIME

Date and time when the event started.

EVENT_END_TIME

Date and time when the event ended.

EVENT_DURATION

Duration of the event. Data Integration Hub calculates the duration based on EVENT_START_TIME and EVENT_END_TIME.

SOURCE_SUCCESS_ROWS

Number of rows that Data Integration Hub read successfully from the source.

SOURCE_FAILED_ROWS

Number of rows that Data Integration Hub failed to read from the source.

TARGET_SUCCESS_ROWS

Number rows that Data Integration Hub read successfully from the target.

TARGET_FAILED_ROWS

Number of rows that Data Integration Hub failed to read from the target.

Note:

- In publications, source refers to the publishing application and target refers to the Data Integration Hub publication repository.
- In subscriptions, source refers to the Data Integration Hub publication repository and target refers to the subscribing application.
- The number of successful source and target rows and the number of failed source and target rows show only for events that were created in the Data Integration Hub repository after the installation of Data Integration Hub version 9.6.2, or after an upgrade of an older version of Data Integration Hub to version 9.6.2.

INDEX

A

architecture
- components 12

automatic mapping
- guidelines 20
- logs 20

B

batch workflow
- definition 41
- process 17

C

catalog API
- description 61
- response 61

catalog API response
- example 63

custom mapping
- description 21
- guidelines 22

custom mappings
- developing 23

D

data extraction APIs
- catalog API 61
- description 61
- events view 64

developing custom mappings
- process 23

developing real-time workflows
- process 42

DX_Add_Document_To_Event
- definition 30
- ports 30
- properties 31

DX_Event_Attribute
- definition 31
- ports 32
- properties 32

DX_Event_Details
- definition 32
- ports 33
- properties 33

DX_Generate_Temporary_File
- definition 33
- ports 34
- properties 34

DX_Notification
- definition 34
- ports 35
- properties 35

DX_Publication_Parameters
- definition 36
- ports 36

DX_Start_Publication
- definition 36
- ports 37
- properties 37

DX_Throw_Error
- definition 38
- ports 38
- properties 39

E

event status API
- process status 54, 55
- response 54, 55

events view
- description 64

example
- catalog API response 63

F

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

O

overview
- description 10

P

PowerCenter
- create mapping 26
- creating source definition 24
- creating target definition 25
- creating the workflow 26
- export workflow 26, 44
- integration 19
- save workflow 26, 44
- sources and targets 24

PowerCenter integration
developing custom mappings 23
publication
process 15, 16
publication repository
datatypes 21

R

real-time workflow
create mapping 43
create topic 44, 45
creating the workflow 44
definition 41
process 17
renaming the workflow target 45
sources and targets 42
real-time workflows
developing 42
guidelines 41
run publication subscription API
command line 60
command line API 58
process status 54, 55
web service 56–58
run publication subscription APIs
description 53
run publication subscription command line api
command syntax 58
run publication subscription command line API
notifications 60
run publication subscription process
status 54, 55
run publication subscription status
event status API 54, 55
run publication subscription web service API
action status 57
description 56
security 58
run publication subscription web service API action
status 57

S

subscription
process 15, 18

T

topic
overview 15
Transformation
DX_Add_Document_To_Event 30
DX_Event_Attribute 31
DX_Event_Details 32
DX_Generate_Temporary_File 33
DX_Notification 34
DX_Publication_Parameters 36
DX_Start_Publication 36
DX_Throw_Error 38
Transformations
configuring 28
definition 27
error handling 28
guidelines 29
installing plug-ins 28

U

user roles
developer 18

W

workflow parameters
customizing 51
workflows
managing 45
properties 46
Workflows
real-time workflow 42