Informatica® Data Integration Hub
10.4.1

Developer Guide

Informatica Data Integration Hub Developer Guide
10.4.1
August 2020

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

NOTICES

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

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2022-03-25

Table of Contents

Preface

Use the *Data Integration Hub Developer Guide* to learn how to create PowerCenter workflows, Data Engineering Integration mappings, Data Quality mappings, and Informatica Cloud tasks to process Data Integration Hub publications and subscriptions. It also provides reports on the data processed by Data Integration Hub.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit https://network.informatica.com.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at http://velocity.informatica.com. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

To find online support resources on the Informatica Network, visit https://network.informatica.com and select the eSupport option.

CHAPTER 1

Introduction to Data Integration Hub

This chapter includes the following topics:

Data Integration Hub Overview

Data Integration Hub is an application integration solution that your organization can use to share and synchronize data between different applications in the organization.

To publish data to Data Integration Hub, first define the data set that you want to manage, for example, sales, customers, or orders. You define a data set by defining a topic. A topic defines the structure of the data that Data Integration Hub stores in the publication repository and the type of publication repository where data is stored. You can manage multiple topics that represent different data sets in Data Integration Hub. Applications publish data to topics and subscribe to data sets that are represented by topics.

Multiple applications can publish to the same topic, for example, different ordering applications can publish their orders to the same Orders topic. Multiple subscribers can consume the data from a topic. Different subscribing applications can consume the data in different formats and in different latencies based on a defined schedule.

Data Integration Hub stores the data that applications publish to topics in the Data Integration Hub publication repository. Data Integration Hub keeps the data in the publication repository until all subscribers consume the data and the retention period expires, and then deletes the data from the publication repository.

Applications can use PowerExchange® adapters and Informatica Intelligent Cloud Services℠ connectors to share data from different sources, such as database tables, files, or any sources that Informatica supports. Each application can be a publisher and a subscriber to different topics.

Publications publish to a specific topic. A publication defines the data source type and the location from where Data Integration Hub retrieves the data that the application publishes. Subscriptions subscribe to one or more topics. A subscription defines the data target type and the location in the subscribing application to where Data Integration Hub sends the published data.

When you create a publication or a subscription, you can choose to use either an automatic Data Integration Hub mapping or a custom Data Integration Hub mapping. Data Integration Hub creates automatic mappings based on the data structure that you define in the topic. Custom Data Integration Hub mappings are based on PowerCenter® workflows, Data Engineering Integration mappings, or Data Integration tasks that the developer creates and maintains for the publication or the subscription.

Data Integration Hub operator uses Enterprise Data Catalog to discover and leverage existing Data Integration Hub objects, and understand their lineage and impact on other entities in the enterprise.

Examples

You run a data center for a major retail chain. The main office has multiple applications. Some of the applications are located on-premises and some are located on the cloud. Each retail branch has a point-of-sale (POS) application and an inventory application. Your applications and branches require the following data:

Customer service applications

Require up-to-date customer order data.

Sales applications

Require up-to-date product sales data.

Marketing application

Requires a weekly deals report.

Accounting application

Requires a monthly deals report.

Branch applications

Require up-to-date inventory and pricing data.

Business Intelligence (BI) application

Requires a weekly report of sales and marketing data and of user interaction data from the corporate website, for the preceding 12 months.

With Data Integration Hub, you can address the following use-cases:

Share product catalog and prices.

You can share product price updates from the sales department with each branch, as follows:

1. Create a Products topic.
2. For the Product Information Management (PIM) application, define a publication that publishes product details and prices to the Products topic and set the schedule to publish the data daily.
3. For each branch application, define a subscription to the Products topic and set the subscription to consume the published data when it is available in Data Integration Hub.

Share daily sales details.

You can share the daily sales details that you receive from the stores with your central sales application and your customer service applications, as follows:

1. Create a Sales topic.
2. For each branch application, define a publication to the Sales topic, and set the schedule to publish daily.

3. For the sales application, define a subscription to the Sales topic, and set the schedule to consume the data when it is published.
4. For the customer service application, define a subscription to the Sales topic, and set the schedule to consume the data once a week.

Share deal details from Salesforce.

You can share deal details from a Salesforce cloud application with the marketing and accounting applications, as follows:

1. Create a Deals topic.
2. For the Salesforce application, define a cloud publication to the Deals topic, and set the schedule to publish weekly.
3. For the marketing application, define a subscription to the Deals topic, and set the schedule to consume the data once a week.
4. For the accounting application, define a subscription to the Deals topic, and set the schedule to consume the data once a month.

Share business intelligence data.

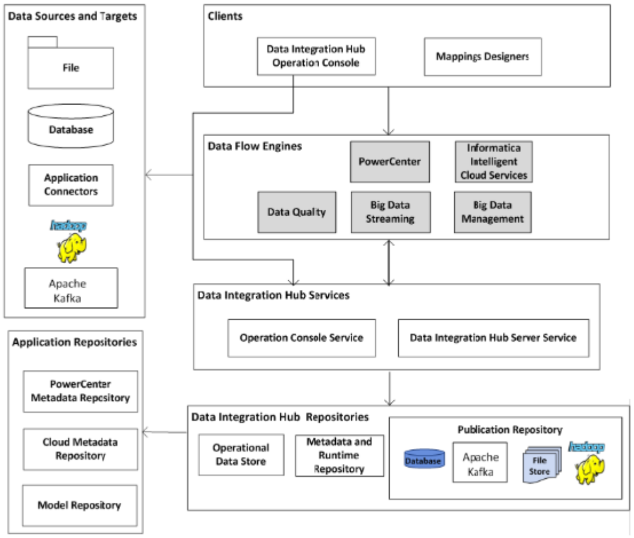
You can share sales data from Salesforce, marketing data from the marketing application, and user interaction data from the corporate website with the BI application as follows:

1. Create a Business Intelligence topic with a Hadoop publication repository and set the publication data retention period to 365 days.
2. For the Salesforce application, define a cloud publication to the Business Intelligence topic, and set the schedule to publish once a week.
3. For the marketing application, define a publication to the Business Intelligence topic, and set the schedule to publish once a week.
4. For the corporate website application, define a cloud publication to the Business Intelligence topic, and set the schedule to publish once a week.
5. For the BI application, define an aggregated subscription to the Business Intelligence topic, and set the schedule to consume the data once a week.

Data Integration Hub Architecture

The Data Integration Hub environment consists of user interface clients, data flow engines, Data Integration Hub services and repositories, and external metadata repositories.

The following image shows the Data Integration Hub components:



Data Integration Hub contains the following components:

Data Integration Hub Operation Console Web client

User interface to manage applications, topics, publications, and subscriptions, and to monitor publications, subscriptions, and events. Administrators also use the Operation Console to manage users and system settings. Developers use the Operation Console to manage Data Integration Hub workflows and connections.

Mappings designer clients

User interfaces to define sources and targets, build custom mappings, and create workflows and tasks. Use the mappings designers if you use custom mappings.

Data flow engines

Engines that retrieve data from publication sources and send the data to subscription targets. You can use different flow engines for different use cases. For example, use PowerCenter to publish and subscribe to on-premises applications, and use Informatica Intelligent Cloud Services to publish and subscribe to cloud applications.

Data Integration Hub Operation Console service

Service that processes actions that users perform on the Operation Console and creates the structure for published data sets in the publication repository.

Data Integration Hub Server service

Service that starts and monitors Data Integration Hub workflows for publications and subscriptions.

Data Integration Hub publication repository

Database that stores published data until the subscribers consume the data. After the data retention period ends, Data Integration Hub deletes the data from the publication repository.

Data Integration Hub metadata repository

Database that stores metadata for Data Integration Hub applications, topics, publications, subscriptions, and events.

Operational data store

A repository that contains aggregated information for reporting purposes. When you install the Data Integration Hub Dashboard and Reports component of Data Integration Hub, Data Integration Hub creates the operational data store repository based on the database connection details that you supply.

PowerCenter metadata repository

Database that stores metadata for PowerCenter mappings, workflows, and transformations.

Cloud metadata repository

Database that stores metadata for cloud mappings and tasks.

Model Repository Service

Database that stores metadata for Data Engineering Integration and Data Quality mappings and transformations.

Data sources and targets

Sources and targets that you use to publish and consume data. You can use the following types of sources and targets:

- Database. Tables and columns.
- File. Binary, text, or unstructured files.
- Application connectors. Connection objects for applications. Available when you use a custom mapping.
- Hadoop. Hadoop Distributed File System (HDFS) and Hive data warehouses.

Data Integration Hub Big Data

Publish and subscribe to high volumes of data, data streams, and data that you want to store for a long period of time with Data Integration Hub. For example, store business intelligence data that you need to review over time on the Data Integration Hub Hadoop publication repository, or publish from and subscribe to Hadoop Distributed File System (HDFS) and Hive data warehouses.

If you want to keep the published data in the Hadoop publication repository after the data is consumed by all subscribers, you can configure Data Integration Hub not to delete published data from the repository.

You can use both automatic mappings and custom mappings to publish and consume big data with Data Integration Hub. For custom mapping publications you can use Informatica Data Engineering Integration mappings and workflows and Informatica Data Engineering Streaming mappings. For custom mapping subscriptions you use Informatica Data Engineering Integration mappings and workflows.

Operation Console

Use the Operation Console user interface to manage applications, topics, publications, and subscriptions, and to monitor publications, subscriptions, and events. Administrators also use the Operation Console to manage users and system settings. Developers use the Operation Console to manage workflows and connections.

You can view the Operation Console in English or in Japanese. You can switch between the display languages.

The Operation Console contains two areas:

Navigator

Use the navigator to navigate between tasks that you can perform in the Operation Console. The navigator shows in the left pane of the Operation Console.

Current page

Main work area in which you perform the tasks that you select in the Navigator. The current page shows in the right pane of the Operation Console.

Changing the Operation Console Language

You can view the Operation Console in English or in Japanese. You can switch between the display languages.

1. In the browser from where you access Data Integration Hub, set the language to the required language.
2. The **Help** link opens the online help in English. To view the Japanese online help access the following URL:

   ```
   http(s)://<host>:<port>/dih-help-ja
   ```

   Where:

   - `<host>` is the host name or the IP address of the Data Integration Hub server.
   - `<port>` is the port number of the Data Integration Hub server.

   For example:

   ```
   https://dih-releases:19443/dih-help-ja/
   ```

Data Integration Hub Topics

A Data Integration Hub topic is an entity that represents a data domain that is published and consumed in Data Integration Hub. A topic defines the canonical data structure and additional data definitions such as the data retention period.

For example, a Sales topic that represents sales data. Applications from all the stores in the organization publish sales data to the Sales topic. The accounting application subscribes to the Sales topic and consumes published sales data from all stores, or, if a filter is applied, from specific stores.

Before you define publications and subscriptions for the data that is published and consumed in Data Integration Hub, you need to define the canonical structure that will hold the data that is published to Data Integration Hub in the Data Integration Hub publication repository. You define the canonical structure when you define the topic. You can define multiple topics that represent different source data sets.

Data Integration Hub Publications and Subscriptions

Publications and subscriptions are entities that define how applications publish data to Data Integration Hub and how applications consume data from Data Integration Hub. Publications publish data to a defined topic and subscriptions subscribe to topics.

Publications and subscriptions control the data flow and the schedule of data publication or data consumption. An application can be a publisher and a subscriber. Multiple applications can publish to the same topic. Multiple applications can consume data from the same topic.

You can use automatic, custom, and modular publications and subscriptions to publish data and to consume data. You can publish from and subscribe to different sources of data. Because the publishing process and the consuming process are completely decoupled, the publishing source and the consuming target do not have to be of the same data type. For example, you can publish data from a file and consume it into a database.

Automatic publications and subscriptions can publish from and subscribe to a relational database, a file, or a cloud application, or over a REST API.

Custom publications and subscriptions can publish from and subscribe to on-premises applications.

Modular publications and subscriptions can publish from and subscribe to cloud applications.

Publication Process

The publication process includes retrieving the data from the publisher, running any associated mappers, such as a mapping or a task, and writing the data to the relevant topic in the Data Integration Hub publication repository. After the publication process ends, subscribers can consume the published data from the publication repository.

The publication process depends on the publication type.

- Automatic publications can run a Data Integration Hub workflow that is based on a PowerCenter batch workflow or run over a REST API.
- Custom publications can either run a Data Integration Hub workflow that is based on a PowerCenter batch workflow, PowerCenter real-time workflow, Data Engineering Integration mapping or workflow, Data Engineering Streaming mapping, or Data Quality mapping or workflow, or run an Informatica Intelligent Cloud Services task.
- Modular publications run an Informatica Intelligent Cloud Services mapping.

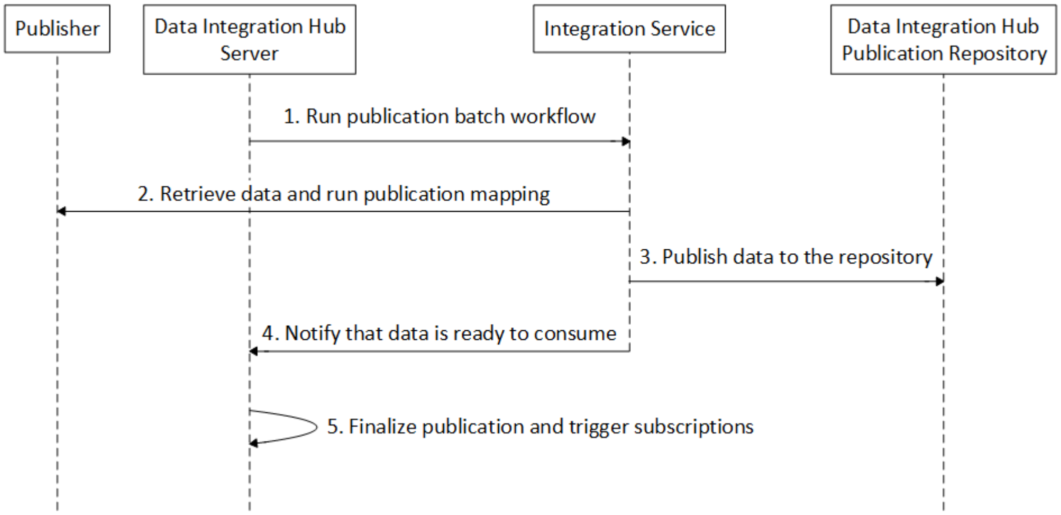
Publication Process with a Batch Workflow

The publication process for publications that run a Data Integration Hub batch workflow includes the following stages:

1. When the publisher is ready to publish the data, the Data Integration Hub server runs the publication batch workflow and sends a request to the relevant Integration Service, either the PowerCenter Integration Service or the Data Integration Service.
2. The Integration Service extracts the data from the publisher and runs the automatic or custom mapping on the data.
3. The Integration Service writes the data to the Data Integration Hub publication repository.
4. The Integration Service notifies the Data Integration Hub server that the published data is ready for subscribers.

5. The Data Integration Hub server changes the status of the publication event to complete and triggers subscription processing.

The following image shows the main stages of the publication process for publications that run a batch workflow:


Publisher
Data Integration Hub Server
Integration Service
Data Integration Hub Publication Repository
1. Run publication batch workflow
2. Retrieve data and run publication mapping
3. Publish data to the repository
4. Notify that data is ready to consume
5. Finalize publication and trigger subscriptions

Publication Process with a Real-time Workflow

The publication process for publications that run a Data Integration Hub real-time workflow includes the following stages:

1. The developer runs the real-time workflow. The workflow writes the data to the relevant tables in the Data Integration Hub publication repository.
2. The Data Integration Hub server triggers a scheduled process and checks for new data in the relevant tables in the Data Integration Hub publication repository.
3. If new data is found, Data Integration Hub updates the publication ID and the publication date of the data to indicate that the data is ready for consumption and creates a publication event in the Data Integration Hub repository.
4. The Data Integration Hub server changes the status of the publication event to complete and triggers subscription processing.

Publication Process with a Data Integration Task

The publication process for publications that run a Data Integration task includes the following stages:

1. When the publication is triggered, either according to schedule or by an external API, the Data Integration Hub server triggers the Data Integration task that is defined for the publication through an Informatica Intelligent Cloud Services REST API.
2. The publication process uses the Data Integration Hub cloud connector to write the data to Data Integration Hub.
3. The Data Integration Hub server changes the status of the publication event to complete and triggers subscription processing.

Publication Process of a Data-driven Publication

The publication process for data-driven publications includes the following stages:

1. After you create a a data-driven publication, you create a POST request to run the publication.
2. When you post the request, Data Integration Hub transfers published data from the request directly to the Data Integration Hub publication repository, to the topic that you define in the publication.
3. Data Integration Hub creates a Data-driven Publication event, based on the event grouping that is defined for the publication:
    - If the grouping time is set to zero, that is, no grouping is defined for the publication, Data Integration Hub creates an event each time data is published to the publication repository.
    - If you define a grouping time, Data Integration Hub creates an event at the end of each grouping period that contains publications. For example, if you configure the publication to group publications every ten seconds, Data Integration Hub creates an event every ten seconds, providing that data was published to the publication repository during the 10-second period.

Subscription Process

The subscription process includes retrieving the required data from the Data Integration Hub subscription repository, running any associated mappers, such as a mapping or a task, and writing the data to one or more subscriber targets. Data Integration Hub keeps the data in the subscription repository until the retention period of the topic expires.

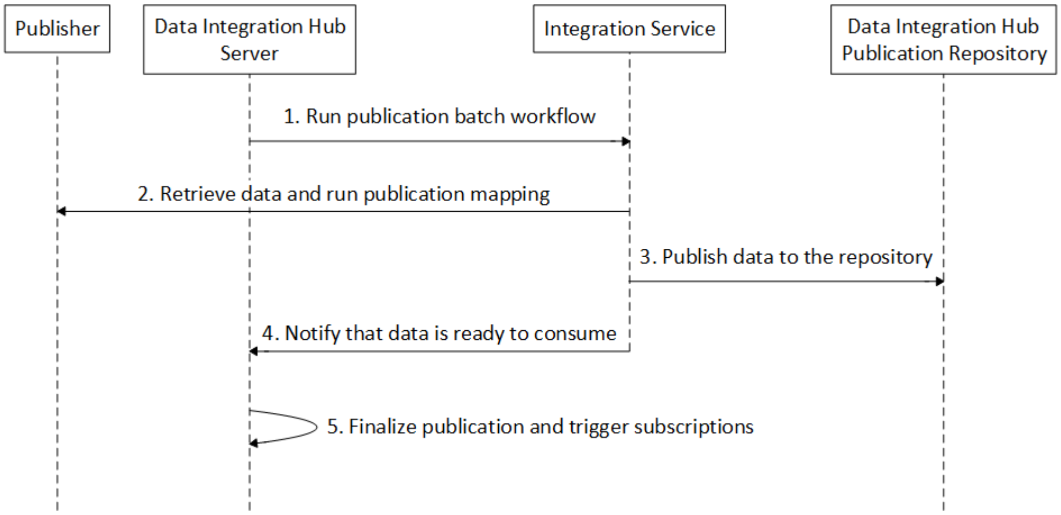
- Automatic subscriptions can run a Data Integration Hub workflow that is based on a PowerCenter batch workflow or run over a REST API.
- Custom subscriptions can either run a Data Integration Hub workflow that is based on a PowerCenter batch workflow, Data Engineering Integration mapping or workflow, Data Engineering Streaming mapping, or Data Quality mapping or workflow, or run an Informatica Intelligent Cloud Services task.
- Modular subscriptions run an Informatica Intelligent Cloud Services mapping.

Subscription Process with a Batch Workflow

The subscription process for subscriptions that run a Data Integration Hub batch workflow includes the following stages:

1. When the publication is ready for subscribers, the Data Integration Hub server runs the subscription batch workflow and sends a request to the relevant Integration Service, either the PowerCenter Integration Service or the Data Integration Service.
2. The Integration Service extracts the data from the Data Integration Hub publication repository, and runs the automatic or custom mapping on the data.
3. The Integration Service sends the required data to the subscriber.
4. The Integration Service notifies the Data Integration Hub server after the subscriber consumed the published data that they require.
5. The Data Integration Hub server changes the status of the subscription event to complete.

The following image shows the main stages of the subscription process for each subscription:


Publisher
Data Integration Hub Server
Integration Service
Data Integration Hub Publication Repository
1. Run publication batch workflow
2. Retrieve data and run publication mapping
3. Publish data to the repository
4. Notify that data is ready to consume
5. Finalize publication and trigger subscriptions

Subscription Process with a Data Integration Task

The subscription process for subscriptions that run a Data Integration task includes the following stages:

1. When the publication is ready for subscribers, the Data Integration Hub server triggers the Data Integration task that is defined for the subscription through an Informatica Intelligent Cloud Services REST API.
2. The subscription process uses the Data Integration Hub cloud connector to read data from Data Integration Hub.
3. The Data Integration task reads the data from Data Integration Hub and then writes the data to the cloud application.
4. The Data Integration Hub server changes the status of the subscription event to complete.

Subscription Process of a Data-driven Subscription

The subscription process for data-driven subscriptions includes the following stages:

1. When you configure the properties of a data-driven subscription, you enter the URL to where Data Integration Hub sends notifications when data is ready to consume from the Data Integration Hub publication repository, from the topic that you define in the subscription.
2. You create a POST request to run the subscription and fetch the data from the Data Integration Hub publication repository, from the topic that you define in the subscription.
3. When Data Integration Hub sends notifications that data is ready to be consumed from the topic, you post the request to run the subscription and to fetch the data.

Developer User Role

The Data Integration Hub developer is responsible for developing workflows, mappings, and tasks for custom publications and subscriptions and for monitoring rules.

Custom publications and subscriptions

The developer can use a PowerCenter workflow, a Data Engineering Integration mapping, Data Quality mapping, or an Informatica Cloud task to define the mappings in custom publications and subscriptions. In publications, the workflow, mapping, or task defines the flow from the publishing application to the Data Integration Hub publication repository. In subscriptions, the workflow, mapping, or task defines the flow from the Data Integration Hub publication repository to the subscribing application.

PowerCenter workflow

The developer creates a workflow in PowerCenter Designer. The developer then creates a Data Integration Hub workflow in the Operation Console and assigns the PowerCenter workflow to the Data Integration Hub workflow. The developer can customize workflow parameters with the Data Integration Hub Forms Designer.

The operator then selects the Data Integration Hub workflow when creating the publication or subscription for an on-premises application.

Data Engineering Integration and Data Engineering Streaming mappings

The developer creates a mapping in the Developer tool. The developer then creates a Data Integration Hub workflow in the Operation Console and assigns the Data Engineering Integration or Data Engineering Streaming mapping to the Data Integration Hub workflow. The developer can customize workflow parameters with the Forms Designer.

The operator then selects the Data Integration Hub workflow when creating a big data publication or subscription.

Data Quality mapping

The developer creates a mapping in the Developer tool. The developer then creates a Data Integration Hub workflow in the Operation Console and assigns the Data Quality mapping to the Data Integration Hub workflow. The developer can customize workflow parameters with the Forms Designer.

The operator then selects the Data Integration Hub workflow when creating the publication or subscription for an on-premises application.

Informatica Cloud task

The developer creates a task in Informatica Cloud. The operator then selects the task when creating the publication or subscription for a cloud application.

Monitoring rules

The developer creates a workflow in PowerCenter Designer. The developer then creates a Data Integration Hub workflow in the Operation Console and assigns the PowerCenter workflow to the Data Integration Hub workflow. The developer can customize workflow parameters with the Forms Designer.

The operator then selects the Data Integration Hub workflow when creating the monitoring rule.

The developer also manages source and target connections for publications and subscriptions in the Operation Console.

CHAPTER 2

PowerCenter Mappings and Workflows

This chapter includes the following topics:

PowerCenter Mappings and Workflows Overview

PowerCenter is a processing engine that Data Integration Hub uses to run Data Integration Hub publications and subscriptions from and to on-premises applications, to run publication pre-processes and subscription post-processes, and to perform actions for monitoring rules that invoke workflows.

For automatic mapping publications and subscriptions, Data Integration Hub creates the PowerCenter mappings that process the publications and subscriptions based on the data structure that you define in the topic. For custom mapping publications and subscriptions, you use the PowerCenter Client tools to develop the PowerCenter mappings and workflows that process the publications and subscriptions. You also develop PowerCenter mappings and workflows for publication pre-processing, subscription post-processing, and monitoring rules that invoke PowerCenter workflows.

After you develop the workflow in PowerCenter, you use the Data Integration Hub Operation Console to import the PowerCenter workflow into a Data Integration Hub workflow. For details, see "Creating a Data Integration Hub Workflow" on page 62.

The Data Integration Hub operator then creates a publication, a subscription, or a monitoring rule in the Data Integration Hub Operation Console, and selects the Data Integration Hub workflow or workflows which are based on the PowerCenter workflow. For more information, see the *Data Integration Hub Operator Guide*.

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

PowerCenter Workflow Types

Data Integration Hub uses PowerCenter batch workflows for publications and subscriptions with automatic mappings.

You can use PowerCenter workflows for publications and subscriptions with custom mappings, publication pre-processing, subscription post-processing, and monitoring rules that invoke PowerCenter workflows. The type of PowerCenter workflow that you can use depends on the type of process for which you want to use the workflow:

PowerCenter Batch Workflows

You can use batch workflows for all processes.

PowerCenter Real-time Workflows

You can use real-time workflows for custom publications.

Batch Workflows

Batch workflows run according to a trigger and not continuously. Data Integration Hub uses batch workflows to process publications and subscriptions, to run publication pre-processes and subscription post-processes, and to perform actions for monitoring rules that invoke workflows. 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 or a request to start a post-processing workflow for a subscription.
- A publication process or a subscription process meets the conditions of a monitoring rule that is configured to invoke a workflow.

When an publication or a subscription 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. When a monitoring rule invokes a workflow, Data Integration Hub does not create an event.

Real-time Workflows

Data Integration Hub uses real-time workflows to process publications. Real-time workflows process data from real-time sources on demand. Real-time sources can be web-service providers, Apache Kafka, or Java Message Service (JMS) queues.

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.

Automatic PowerCenter Mappings and Workflows

When you create a publication or a subscription with an automatic mapping, Data Integration Hub creates PowerCenter components that process the publication or subscription according to the definitions that you set in the Publication or Subscription wizard.

Data Integration Hub creates the following PowerCenter entities for a publication or a subscription with an automatic mapping:

- Batch workflow. Scheduled batch workflow that the Data Integration Hub triggers to process the data.
- Data processing session. Session with transformations that extract the data from the publication source or write the data to the subscription target. The session also includes a mapping, source and target objects.
- Reporting session. Session that sends and receives notifications about the publication or subscription process. This session updates the publication or subscription events after processing. The session also includes a mapping, source and target objects.
- Metadata folders in the PowerCenter repository. The folders contain workflow, mapping, and session information. A separate folder contains publication or subscription metadata. When you create an automatic publication mapping, PowerCenter creates subscription source shortcuts based on the publication target.

Automatic PowerCenter Mappings and Workflows Rules and Guidelines

When you publish or consume data sets with automatic mappings, consider the following rules and guidelines:

- Data Integration Hub creates PowerCenter entities for each automatic mapping. Each PowerCenter entity name starts with the prefix **DIH__**. Do not manually change or delete the PowerCenter entities.
- You cannot use high precision processing in PowerCenter to run automatic mappings. To use high precision, use a custom mapping.
- Do not manually change or delete tables in the publication repository that store data sets from publications with automatic mappings.
- Automatic mappings run with the DIH__STAGING and the DIH__REPO PowerCenter connections. To prevent publication or subscription failure, do not change or delete the connections.
- When you publish varbinary or nvarchar datatypes from a Microsoft SQL Server database source, PowerCenter converts the data size limit to a fixed size of 100 MB. If you create a subscription with an automatic mapping that writes the published data to a Microsoft SQL Server database target, you must change the data size limit to MAX when you create the target tables before you run the subscription workflow.
- When you use tables from a default schema to publish from a relational database or to subscribe to a relational database, the schema name is not persistent.
- When you use tables from a non default schema to publish from a relational database or to subscribe to a relational database, the schema name is persistent.
- Schema names are saved in the Mapping tab of the PowerCenter session in the following locations:
  - For publications, the schema name is saved in the Source properties in the **Owner Name** field.
  - For subscriptions, the schema name is saved in the Target properties in the **Target Name Prefix** field.

Automatic PowerCenter Mappings and Workflows Logs

When you publish or consume data sets with automatic mappings, the PowerCenter Integration Service creates the following logs:

Publication workflow logs

The PowerCenter Integration Service creates publication workflows with the following settings:

Parameter	Value
Save Workflow Log By	Runs
Save Workflow Log for These Runs	$PMWorkflowLogCount

The value of *$PMWorkflowLogCount* in the PowerCenter Integration Service determines the number of logs that the Integration Service creates for each workflow run.

Publication session logs

The PowerCenter Integration Service creates publication workflow sessions with the following settings:

Parameter	Value
Save Session Log By	Runs
Save Session Log for These Runs	$PMSessionLogCount

The value of *$PMSessionLogCount* in the PowerCenter Integration Service determines the number of logs that the Integration Service creates for each session run.

Subscription logs

The PowerCenter Integration Service creates subscription workflows with a concurrent run set. The number of logs is unlimited with timestamps.

Custom PowerCenter Mappings and Workflows

You can develop PowerCenter workflows and import them to Data Integration Hub for the following use-cases:

- You need to prepare data before you publish it from an on-premises application.
- You require additional processing or transformation to the data set to publish from or to consume into an on-premises application.
- You need to add logic to data to which you subscribe, for either an on-premises or a cloud application.
- You create a monitoring rule that invokes a workflow when a publication process or a subscription process meets the conditions of the rule.

You develop the workflows for these use-cases in a similar way that you develop other PowerCenter workflows. The operator then selects the workflow to use in a publication or a subscription with a custom mapping, in a publication pre-process or a subscription post-process, or in a monitoring rule, as applicable.

Supported Datatypes

When you publish data sets to the publication repository with a PowerCenter workflow, you can publish datatypes based on the database type of the publication repository.

The following table describes the supported datatypes you can publish to the publication repository:

Database Type	Datatypes
Oracle	- blob - clob - number - number($p, $s) - timestamp up to 29 bit - varchar2($l char)
Microsoft SQL Server	- bigint - datetime - decimal($p,$s) - float - int - nvarchar($l) - nvarchar(MAX) - varbinary(MAX)

Custom PowerCenter Mappings and Workflows Rules and Guidelines

When you develop PowerCenter workflows to use in publications and subscriptions with custom mappings, publication pre-processes, subscription post-processes, and monitoring rules, consider the following rules and guidelines:

General rules and guidelines

- Before you develop workflows for Data Integration Hub in PowerCenter, verify that the Data Integration Hub PowerCenter client and server plug-ins are installed and registered to the PowerCenter repository. For details, see the *Data Integration Hub Installation and Configuration* Guide.
- Name and store PowerCenter entities for custom mappings in different folders with a different naming convention from the naming convention of PowerCenter entities for custom mappings.
- Data Integration Hub uses a separate reporting session to update the status of publication and of subscription events that use an automatic mapping. It is recommended to create separate sessions for data processing and reporting, similar to automatic mappings. You can use a workflow for an automatic mapping as a reference.
- You can use user-defined session parameters in custom workflows and define their values in Data Integration Hub or in a parameter file. You can manage the values of user-defined session parameters in Data Integration Hub in the Forms Designer. You cannot manage the values of built-in session parameters in Data Integration Hub. For more information about session parameters, see the section "Working with Session Parameters" in the *PowerCenter Advanced Workflow Guide*.

  Note: Data Integration Hub does not support session parameters in the format `$InputFile_$$CustomVariable`.
- Informatica recommends that custom publications and subscriptions that use PowerCenter mappings use the same event statuses and types as those that Data Integration Hub assigns to automatic mappings. For details, see the *Data Integration Hub Operator Guide*. You assign event statuses and types in the DX_Event_Details transformation.

- 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 and subscriptions with automatic mappings.
- If you publish from a database source or write to a database target with a different database type from the publication repository database type, Data Integration Hub converts the data to a data type that the publication repository database supports. Therefore, if you consume the published data from the publication repository to a different target database, verify that the data type conversion does not create run-time errors during processing. For example, if you publish data from a Microsoft SQL Server database source to an Oracle publication repository, and then consume the published data to a Microsoft SQL Server database target, MIN or MAX values might be converted to a value that is higher or lower than values that the Microsoft SQL Server database target supports.
- 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 “Creating a Data Integration Hub Workflow” on page 62.
- For additional rules and guidelines that apply to batch workflows, see “Developing PowerCenter Batch Workflows Rules and Guidelines” on page 29.
- For additional rules and guidelines that apply to real-time workflows, see “Developing PowerCenter Real-time Workflows Rules and Guidelines” on page 32.

Publication rules and guidelines

- If you publish from a database source, you cannot use the following special characters in table names and in column names of a publication target: space ( ), dash (-), and period (.). The publication process replaces the characters with underscores (_).
- When you develop a publication pre-process workflow, call the DX_Start_Publication transformation at the end of the pre-processing workflow, for example in a separate mapping. The transformation instructs the Data Integration Hub server to trigger the publication process. When you configure the DX_Start_Publication transformation, consider the following guidelines:
  - When a publication pre-process starts a single publication, use the DXEventId port. The event ID 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.

    Note: If you do not define a DXEventId port you must define a DXPublicationName port.
  - 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: 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.
- When you develop a workflow for a publication with a file source, if the path of the source file is parameterized, Data Integration Hub picks up the file and moves it to the Data Integration Hub document store. If the path of the source file is hard coded, a PowerCenter source picks up and processes the file. For source files with a parameterized file path, the following rules apply:
  - For flat file sources, the source file type must be indirect.
  - For pass-through file sources, the source file type must be direct.
- When you select a Data Integration Hub workflow that is based on a PowerCenter workflow to use in a publication with a custom mapping, Data Integration Hub creates the structure of the published data set in the publication repository based on the target definitions of the workflow.

Subscription rules and guidelines

- When you develop a subscription post-processing workflow, call the DX_Notification transformation at the end of the workflow. You can find a sample post-processing workflow in the following directory: `<DIHInstallationDir>/samples/post_processing_workflow`.
- When you develop a workflow for a compound subscription, define the behavior if the compound subscription starts manually before all published data sets are ready to consume. For example, you can instruct the mapping to fail the workflow or to ignore empty tables. Published data sets that are not ready to consume have the publication instance ID 0.
- When you develop a workflow for a subscription with a file target, you can parameterize the target file path. The following rules and guidelines apply when you parameterize the file path:
  - For flat file targets, the target file parameter must start with `$OutputFile`.
  - For pass-through file targets, the target file parameter must start with `$OutputFile_DIHRepoFile_`.
  - When the Data Integration Hub operator creates the subscription in the Data Integration Hub Operation Console, they must specify the target output file name as the value for the output file parameter.
  - The value of the output file parameter can contain a pattern that ensures that the name is unique for each file, for example `($sequence)`.
- When you develop a workflow for a subscription that consumes data from topic tables where delta detection is applied, add Update Strategy transformations to the mapping, and define the update strategy for data that exists in the target application. Add one of the following flags for each row in topic tables where delta detection is applied:

Flag	Operation
I	Insert row
U	Update row
D	Delete row
N	No operation

Developing PowerCenter Batch Workflows

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

Developing PowerCenter Batch Workflows Rules and Guidelines

When you develop PowerCenter batch workflows to use in publications and subscriptions with custom mappings and in monitoring rules, consider the following rules and guidelines:

- When you add PowerCenter transformations to a PowerCenter mapping, you can add workflow parameters that the corresponding Data Integration Hub workflow can use. You can use the Forms Designer, in the Data Integration Hub Operation Console, to customize the layout and behavior of the workflow parameters that appear when the operator creates or edits a publication or a subscription with a custom mapping.
- If you edit the PowerCenter workflow or mapping, you must update the associated Data Integration Hub workflow by re-selecting the PowerCenter workflow or the XML file in the Data Integration Hub Operation Console. During the update process, you can resolve changes to workflow parameters and parameter types. When you change the type of a workflow parameter type, the import process deletes the values for the imported parameters and you must manually enter the parameter values.

For general rules and guidelines on developing PowerCenter workflows to use with Data Integration Hub, see Custom PowerCenter Mappings and Workflows Rules and Guidelines.

Developing PowerCenter Batch Workflows for Publications and Subscriptions Process

To develop a batch workflow in PowerCenter that processes Data Integration Hub data, perform the following steps:

1. Create the source and target definitions.
2. Create the mapping and add transformations.
3. Create the workflow and the session.
4. Save the PowerCenter workflow to the PowerCenter repository.

Note: This process is not applicable for monitoring rule workflows. There are no limitations on how you define workflows for monitoring rules.

Step 1. Create the Source and Target Definitions

When you develop a publication workflow, you create the source based on the structure of the database or file that you want to publish and set the target to the publication repository.

If you publish from a database source, verify that you publish datatypes that the publication repository can store.

When you develop a subscription workflow, you create a source based on the topic structure from the publication repository. You can copy the source from the publication metadata folder in the PowerCenter repository.

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

Creating the Source Definition for a Subscription Workflow

In the PowerCenter Designer, create the source and define the source properties of a subscription workflow. When you develop a publication workflow, you define the source based on the application from which you want to publish data.

The source definition process includes the following steps:

1. Create the source object. Set the source connection to DIH__STAGING. You create a source based on the topic structure from the publication repository. You can copy the source from the publication metadata folder in the PowerCenter repository.
2. Add variables to filter the published data to consume. For example, you can select to consume data from a specific table in the publication repository.
3. Add the required fields that determine the data set to consume. You can define multiple data sets to consume, similar to an aggregated or a compound subscription with an automatic mapping.

The following table describes the fields to add to the source object of a subscription workflow:

Field	Description
DIH__PUBLICATION_INSTANCE_ID	Required. Identifiers of one or more published data sets in a comma-separated list. Each data set that an application publishes has a unique identifier. To filter the data to consume, use the value from the **$$<topicName>__PublicationInstanceIDs** workflow parameter. The parameter datatype must be number(19) if you write to an Oracle database target or number(19,0) if you write to a Microsoft SQL Server database target.

DIH__PUBLICATION_INSTANCE_DATE	Date and time that each application started publishing the data sets, in a comma-separated list. If you use database partitions, you can filter the data to consume by using the value from the **$$<topic_name>__PublicationInstanceDatesSQL** workflow parameter. The value format depends on the publication repository database type.
	On an Oracle database, the datatype must be date and the value must be in the following format:
	`YYYY-MM-DD HH24:MI:SS`
	On a Microsoft SQL Server database, the datatype must be datetime and the value must be in the following format:
	`yyyy-mm-dd hh:mi:ss (24h)`
	Note: If you want to filter the data to consume with a different transformation, you can use the **$$<topic_name>__PublicationInstanceDates** parameter instead.

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

```
MY_TABLE.DIH__PUBLICATION_INSTANCE_ID in ($$myTopic__PublicationInstanceIDs)
AND MY_TABLE.DIH__PUBLICATION_INSTANCE_DATE in ($$myTopic__PublicationInstanceDatesSQL)
```

Creating the Target Definition

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

The target definition process includes the following steps:

1. Create the target object. If you develop a publication workflow, use the DIH__STAGING connection in the target definition. The database target type must match the publication repository database type.
2. If you develop a publication workflow, add the required fields that determine the data set to publish.

The following table describes the required fields to add to the target object of a publication:

Parameter	Description
DIH__PUBLICATION_INSTANCE_ID	Required. Identifier of the published data set. Each data set that an application publishes has a unique identifier. The field uses the value from the $$publicationInstanceID workflow parameter.

DIH__PUBLICATION_INSTANCE_DATE	Required. Date and time that the application started publishing the data set. The field uses the value from the $$publicationInstanceDate workflow parameter. The value format depends on the publication repository database type.
	On an Oracle database, the value must be in the following format:
	`to_date('<DATE>', 'YYYY-MM-DD HH24:MI:SS')`
	On a Microsoft SQL Server database, the value must be in the following format:
	`convert(datetime,'<DATE>',120)`

Step 2. Create the Mapping

Create a mapping that contains the source definition, target definition, and transformations that you want to use to process data for Data Integration Hub.

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

Step 3. Create the PowerCenter Workflow and Session

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

1. In the Workflow Designer, create the workflow and the session object. For general information about creating workflows and sessions, see the *PowerCenter Advanced Workflow Guide*.
2. Add the transformations that notify the Data Integration Hub server that the publication or that the subscription process ended.
    - If you develop a publication workflow, add the DX_Notification transformation to trigger the subscription workflows and change the publication event status.
    - If you develop a subscription workflow, add the DX_Event_Details to change the subscription event status.

3. Save the session object.
4. Test the workflow to ensure that it works correctly.

Step 4. Save the PowerCenter Workflow

Save the workflow to the PowerCenter repository. After you save the workflow, you can export it 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`.

Developing PowerCenter 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.

Developing PowerCenter 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*.
- If you edit the PowerCenter workflow or mapping, you must update the associated Data Integration Hub workflow by re-selecting the PowerCenter workflow or the XML file in the Data Integration Hub Operation Console.

For general rules and guidelines on developing PowerCenter workflows to use with Data Integration Hub, see Custom PowerCenter Mappings and Workflows Rules and Guidelines.

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 real-time data that you want to publish. You set the target of the publication real-time workflow to the Data Integration Hub publication repository. You can also add properties to store other Data Integration Hub values that you want to send from PowerCenter.

You define the source of a publication real-time workflow based on the following real-time data:

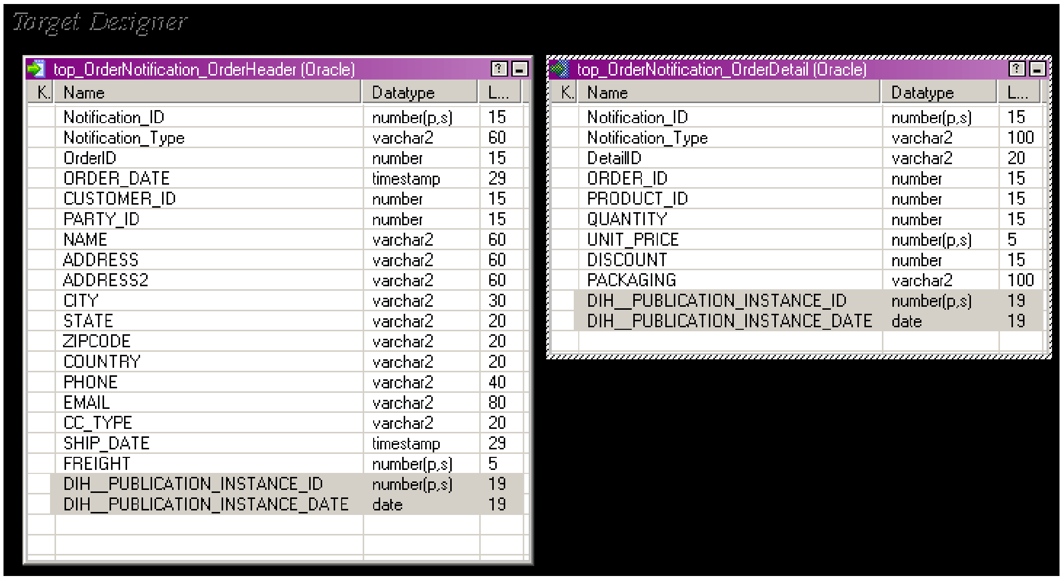
- Data in the Apache Kafka stream
- Structure of the JMS
- Web service source

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:



top_OrderNotification_OrderHeader (Oracle)

K.	Name	Datatype	L...
	Notification_ID	number(p,s)	15
	Notification_Type	varchar2	60
	OrderID	number	15
	ORDER_DATE	timestamp	29
	CUSTOMER_ID	number	15
	PARTY_ID	number	15
	NAME	varchar2	60
	ADDRESS	varchar2	60
	ADDRESS2	varchar2	60
	CITY	varchar2	30
	STATE	varchar2	20
	ZIPCODE	varchar2	20
	COUNTRY	varchar2	20
	PHONE	varchar2	40
	EMAIL	varchar2	80
	CC_TYPE	varchar2	20
	SHIP_DATE	timestamp	29
	FREIGHT	number(p,s)	5
	DIH__PUBLICATION_INSTANCE_ID	number(p,s)	19
	DIH__PUBLICATION_INSTANCE_DATE	date	19

top_OrderNotification_OrderDetail (Oracle)

K.	Name	Datatype	L...
	Notification_ID	number(p,s)	15
	Notification_Type	varchar2	100
	DetailID	varchar2	20
	ORDER_ID	number	15
	PRODUCT_ID	number	15
	QUANTITY	number	15
	UNIT_PRICE	number(p,s)	5
	DISCOUNT	number	15
	PACKAGING	varchar2	100
	DIH__PUBLICATION_INSTANCE_ID	number(p,s)	19
	DIH__PUBLICATION_INSTANCE_DATE	date	19

Step 2. Create the Mapping

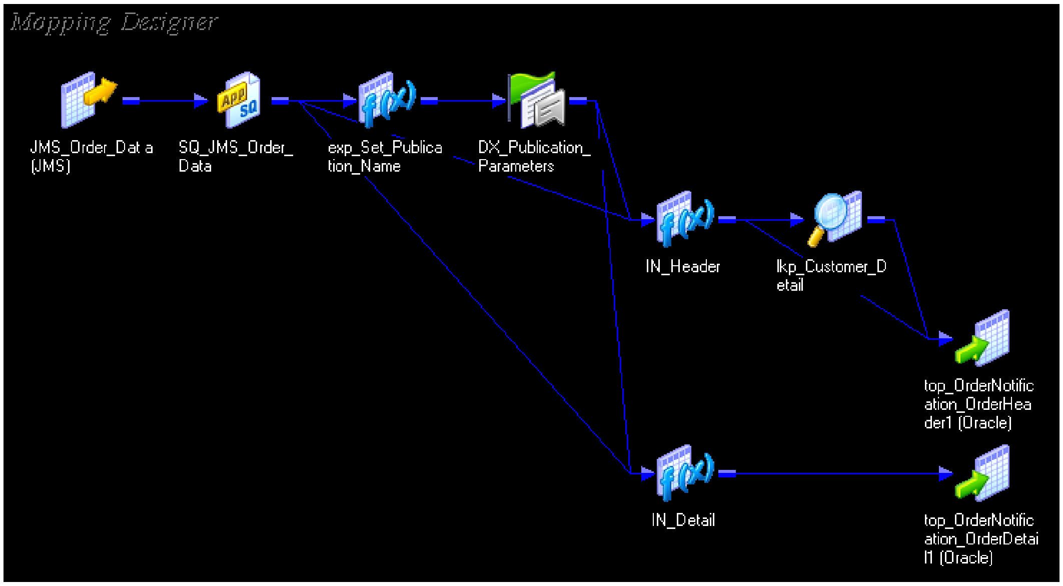
Create a mapping that contains the source definition, the target definition, and the transformations that you want to use in the real-time publication workflow.

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:


Mapping Designer
JMS_Order_Dat a (JMS)
SQ_JMS_Order_ Data
exp_Set_Publica tion_Name
DX_Publication_ Parameters
IN_Header
lkp_Customer_D etail
top_OrderNotific ation_OrderHea der1 (Oracle)
IN_Detail
top_OrderNotific ation_OrderDetai l1 (Oracle)

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.

CHAPTER 3

Data Engineering Integration and Streaming Mapping and Workflows

This chapter includes the following topics:

Data Engineering Integration and Streaming Mapping and Workflows Overview

Data Integration Hub uses Data Engineering Integration and Data Engineering Streaming to run Data Integration Hub big data publications and subscriptions.

You use Data Engineering Integration mappings to run custom batch publications and subscriptions that publish and consume large, diverse, and fast changing data sets. You use Data Engineering Integration workflows with multiple mappings in a workflow to run multiple custom batch publications and subscriptions that publish and consume large, diverse, and fast changing data sets. You use Data Engineering Streaming mappings to run custom multi-latency publications that publish streams of data in real time.

You create a Data Engineering Integration workflow by using multiple Data Engineering Integration mappings.

The Data Integration Service runs the Data Engineering Integration mapping, Data Engineering Streaming mappings, Data Engineering Integration workflows on the Hadoop environment.

You use the Developer tool to develop the Data Engineering Integration, Data Engineering Streaming mappings, and Data Engineering Integration Workflow that process the publications and subscriptions. You then use the Data Integration Hub Operation Console to import the mappings into a Data Integration Hub workflow. For details, see "Creating a Data Integration Hub Workflow" on page 62.

The Data Integration Hub operator creates a publication or a subscription in the Data Integration Hub Operation Console, and selects the Data Integration Hub workflow which is based on the Data Engineering Integration, Data Engineering Streaming mapping, or Data Engineering Integration Workflow. For more information, see the *Data Integration Hub Operator Guide*.

Sample mappings

You can find sample mappings in the following locations:

- Data Engineering Integration mappings: `<DIHInstallationDir>/samples/bdm_mappings`. Each sample mapping has an associated readme file that describes the sample mapping and contains guidelines for using the mapping as a basis to create your own mappings.
- Data Engineering Streaming mappings: `<DIHInstallationDir>/samples/bds_mappings`. Under this folder, there are sub-folders with sample mappings for an Oracle publication repository and for a Microsoft SQL Server publication repository. The readme file that resides in this folder describes the sample mappings and contains guidelines for using the mappings as a basis to create your own mappings.
- Data Engineering Integration Workflow: `<DIHInstallationDir>/samples/bdm_workflows`. Under this folder, there are sub-folders with sample workflows for an Oracle publication repository and for a Microsoft SQL Server publication repository. The readme file that resides in this folder describes the sample workflows and contains guidelines for using the workflows as a basis to create your own workflows.

Data Engineering Integration and Streaming Mappings and Workflows in Data Integration Hub

You use Data Engineering Integration mappings and workflows in Data Integration Hub publications and subscriptions with custom batch mappings. You use Data Engineering Streaming mappings and workflows in Data Integration Hub publications with custom multi-latency mappings.

In publication mappings, the source is the publishing application and the target is the Data Integration Hub publication repository. In subscription mappings, the source is the Data Integration Hub publication repository, and the target is the subscribing application.

You can use user-defined mapping parameters in custom batch mappings and define their values in Data Integration Hub or in a parameter file. You can manage the values of mapping parameters based on developer tool mappings. You cannot manage the values of built-in mapping parameters in Data Integration Hub. For more information about mapping parameters, see the *Informatica Developer Tool Guide*.

Note: Data Integration Hub does not support session parameters in the following formats: `$InputFile_`, `$OutputFile_`.

Before You Begin

Before you develop Data Engineering Integration, Data Engineering Streaming mappings, or Data Engineering Integration workflows to use in Data Integration Hub, verify that the following conditions exist:

- The following Data Integration Hub components are installed:
  - Data Integration Hub Hadoop Service.
  - Data Integration Hub Data Engineering Integration.
- The topic to which to publish data and from which to consume data is configured in Data Integration Hub. The topic must be a Hadoop-based topic.
- The following connections are configured in Data Integration Hub:
  - Connection to the publishing application.
  - Connection to the subscribing application.
- A Hadoop connection to where to run the mappings exists in your environment. The connection must be a cluster connection, and must push mapping logic to the Hadoop cluster.

Before you develop Data Engineering Streaming mappings, copy the runtime `.jar` files from `<DIHInstallationDir>/ powercenter/lib` to the following location:

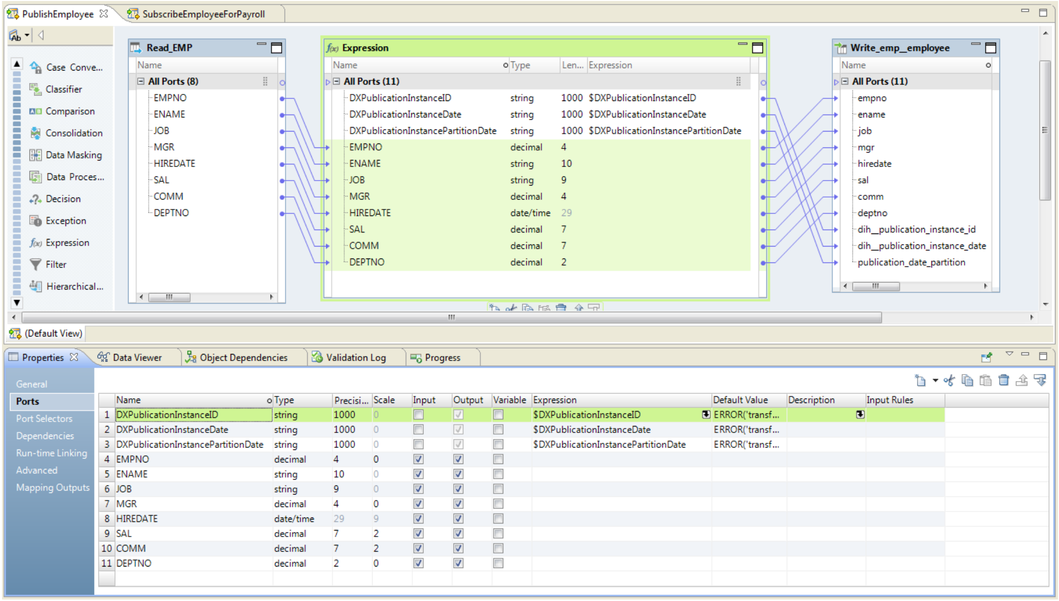
```
BDS_HOME/services/shared/hadoop/$Distribution/extras/spark-auxjars/
```

Developing Data Engineering Integration Mappings for Publications

To develop a Data Engineering Integration mapping for a publication, perform the following steps in the Developer tool:

1. Create source and target connections. The source connection is a connection to the publishing application and the target connection is a Hive connection to the Data Integration Hub publication repository.
2. Create source and target data objects.
3. Create a mapping and add the source and target objects to the mapping.
4. Add Data Integration Hub parameters to the mapping.
5. Add an Expression transformation to the mapping, configure ports in the transformation, and connect ports between the source, the transformation, and the target.
6. Configure the mapping run-time environment and create an application from the mapping.

The following image shows a sample publication mapping:



	Name	Type	Precisi...	Scale	Input	Output	Variable	Expression	Default Value	Description	Input Rules
1	DXPublicationInstanceID	string	1000	0	□	☑	□	$DXPublicationInstanceID	ERROR('transf...		
2	DXPublicationInstanceDate	string	1000	0	□	☑	□	$DXPublicationInstanceDate	ERROR('transf...		
3	DXPublicationInstancePartitionDate	string	1000	0	□	☑	□	$DXPublicationInstancePartitionDate	ERROR('transf...		
4	EMPNO	decimal	4	0	☑	☑	□				
5	ENAME	string	10	0	☑	☑	□				
6	JOB	string	9	0	☑	☑	□				
7	MGR	decimal	4	0	☑	☑	□				
8	HIREDATE	date/time	29	9	☑	☑	□				
9	SAL	decimal	7	2	☑	☑	□				
10	COMM	decimal	7	2	☑	☑	□				
11	DEPTNO	decimal	2	0	☑	☑	□				

Step 1. Create Source and Target Connections

1. Create a Streaming source connection to the publishing application.
2. Create a target connection to the Data Integration Hub publication repository.

Step 2. Create Source and Target Data Objects

Create data objects under **Physical Data Objects**.

1. Create a source data object and define the column projection in the source connection to publish.
2. Create a target data object and select the table in the target connection to where to publish the data from the source. The object must be a relational data object.

Step 3. Create a Mapping with Source and Target

1. Create and name a new mapping.
2. Add the source physical data object to the mapping as a Reader.
3. Add the target physical data object to the mapping as a Writer.

Step 4. Add Data Integration Hub Parameters to the Mapping

▶ Add the following parameters to the mapping:

```
DXPublicationInstanceID
DXPublicationInstanceDate
DXPublicationInstancePartitionDate
```

Step 5. Add an Expression Transformation to the Mapping

1. Add an Expression transformation to the mapping, between the source and target objects.

2. Link all the ports from the source object to the identical ports in the Expression transformation. For example, if your topic table includes the column `EMPNO`, link the `EMPNO` port in the source object to the `EMPNO` port in the Expression transformation.
3. Configure the following additional ports in the Expression transformation:

   ```
   DXPublicationInstanceID
   DXPublicationInstanceDate
   DXPublicationInstancePartitionDate
   ```

   The data type of all ports is string, and the minimum precision is 200.
4. Link all the ports from the Expression transformation to the identical ports in the target object. For example, link the `EMPNO` port in the Expression transformation to the `EMPNO` port in the target object, and link the `DXPublicationInstanceID` port in the Expression transformation to the `dih__PublicationInstanceID` port in the target object.
5. Save the mapping.

Step 6. Configure the Mapping Run-time Environment and Create an Application

1. In the **Properties** pane select **Run-time**, and then, under **Validation Environments**, select **Hadoop** and then select **Hive**. Verify that **Native** is not selected.
2. Create an application from the mapping.

   The mapping is deployed to the Data Integration Service for Hadoop environment.

Developing Data Engineering Streaming Mappings for Publications

To develop a Data Engineering Streaming mapping for a publication, perform the following steps in the Developer tool:

1. Create source and target connections. The source connection is a connection to the publishing application and the target connection is a connection to the Data Integration Hub publication repository.
2. Create source and target data objects.
3. Create a mapping and add the source and target objects to the mapping.
4. Add an Expression transformation to the mapping, configure ports in the transformation, and connect ports between the source and the transformation.
5. Add a Java transformation to the mapping and map fields from the Expression transformation to the Java transformation and from the Java transformation to the target.
6. Configure the mapping run-time environment and create an application from the mapping.

Step 1. Create Source and Target Connections

1. Create a Streaming source connection to the publishing application.
2. Create a target connection to the Data Integration Hub publication repository.

Step 2. Create Source and Target Data Objects

Create data objects under **Physical Data Objects**.

1. Create a source data object and define the column projection in the source connection to publish.
2. Create a target data object and select the table in the target connection to where to publish the data from the source. The object must be a relational data object.

Step 3. Create a Mapping with Source and Target

1. Create and name a new mapping.
2. Add the source physical data object to the mapping as a Reader.
3. Add the target physical data object to the mapping as a Writer.

Step 4. Add an Expression Transformation to the Mapping

1. Add an Expression transformation to the mapping, between the source and target objects.
2. Link all the ports from the source object to the identical ports in the Expression transformation.
3. Configure the following additional ports in the Expression transformation:

Port	Description
PublicationName	Name of the publication.
DX_SERVER_URL	A valid Data Integration Hub RMI URL. For example: `rmi://localhost:18095`.

Step 5. Add a Java Transformation to the Mapping

1. Copy the Java transformation from the sample mapping. You can find sample mappings in the following location: `<DIHInstallationDir>/samples/bds_mappings`. Under this folder, there are sub-folders with sample mappings for an Oracle publication repository and for a Microsoft SQL Server publication repository.
2. Map the following fields from the Expression transformation to the Java transformation:
   - PublicationName
   - DX_SERVER_URL
3. Map the following fields from the Java transformation to the topic table in the target object:

Java Transformation Field	Topic Table Field
DXPublicationInstanceID	DIH__PUBLICATION_INSTANCE_ID
DXPublicationInstanceDate	DIH__PUBLICATION_INSTANCE_DATE

4. In the target transformation, open the target data object and change the data type of `DIH__PUBLICATION_INSTANCE_ID` from `decimal` to `bigint` .
5. Save the mapping.

Step 6. Configure the Mapping Run-time Environment and Create an Application

1. In the **Properties** pane select **Run-time**, and then, under **Validation Environments**, select **Hadoop** and then select **Spark**. Verify that **Native** is not selected.
2. Create an application from the mapping.

   The mapping is deployed to the Data Integration Service for Hadoop environment.

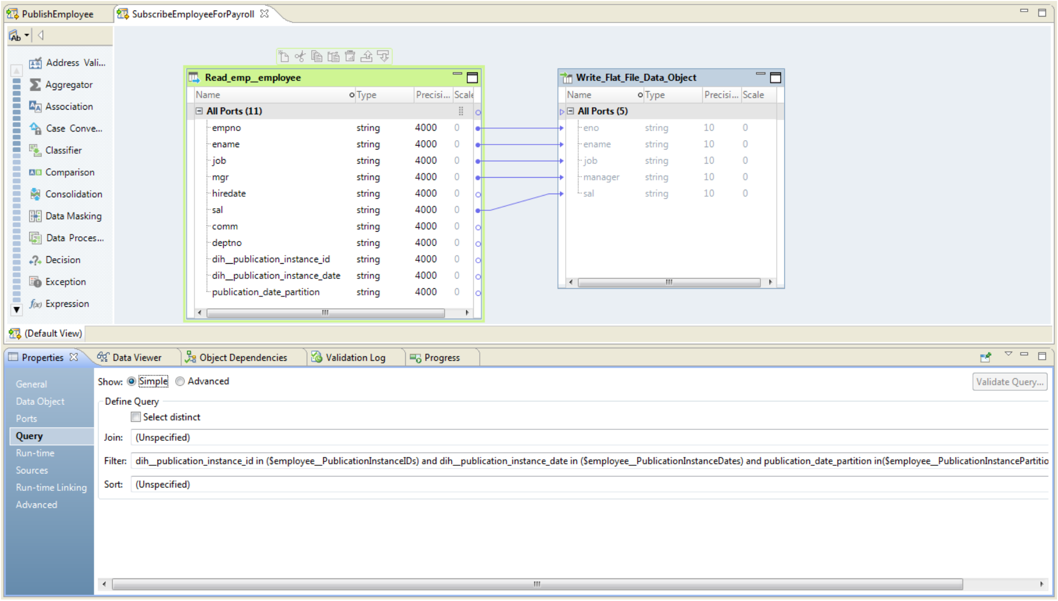
Developing Data Engineering Integration Mappings for Subscriptions

To develop a Data Engineering Integration mapping for a subscription, perform the following steps in the Developer tool:

1. Create source and target connections. The source connection is a Hive connection to the Data Integration Hub publication repository and the target connection is a connection to the subscribing application.
2. Create source and target data objects.
3. Create a mapping, add the source and target objects to the mapping, and connect ports between the source and the target.
4. Add Data Integration Hub parameters to the mapping.
5. Add a Filter query to the source. You can filter subscriptions by publication instance ID, publication date, and publication date partition.

   Note: If you are creating a mapping for an unbound subscription you do not need add a filter query to the mapping.
6. Configure the mapping run-time environment and create an application from the mapping.

The following image shows a sample subscription mapping:



Read_emp__employee

Name	Type	Precisi...	Scale
All Ports (11)			
empno	string	4000	0
ename	string	4000	0
job	string	4000	0
mgr	string	4000	0
hiredate	string	4000	0
sal	string	4000	0
comm	string	4000	0
deptno	string	4000	0
dih__publication_instance_id	string	4000	0
dih__publication_instance_date	string	4000	0
publication_date_partition	string	4000	0

Write_Flat_File_Data_Object

Name	Type	Precisi...	Scale
All Ports (5)			
eno	string	10	0
ename	string	10	0
job	string	10	0
manager	string	10	0
sal	string	10	0

Show: ● Simple ○ Advanced

Define Query

□ Select distinct

Join: (Unspecified)

Filter: dih__publication_instance_id in ($employee__PublicationInstanceIDs) and dih__publication_instance_date in ($employee__PublicationInstanceDates) and publication_date_partition in($employee__PublicationInstancePartitio

Sort: (Unspecified)

Step 1. Create Source and Target Connections

1. Create a Hive source connection to the Data Integration Hub publication repository. Hive must belong to the cluster where the mappings run.
2. Create an Oracle or a Flat File target connection to the subscribing application.

Step 2. Create Source and Target Data Objects

Create data objects under **Physical Data Objects**.

1. Create a source data object and select the table in the source connection to consume. The object must be a relational Hive data object.
2. Create a target data object and select the table in the target connection subscribes to the data. The object can be a relational data object or a flat file data object.

Step 3. Create a Mapping with Source and Target

1. Create and name a new mapping.
2. Add the source physical data object to the mapping as a Reader.
3. Add the target physical data object to the mapping as a Writer.
4. Link all the ports from the source object to the identical ports in the target object. For example, if your topic table includes the column `ENAME`, link the `ENAME` port in the source object to the `ENAME` port in the target object.

Step 4. Add Data Integration Hub Parameters to the Mapping

▶ Add the following parameters to the mapping:

```
<TOPIC_NAME>__DXPublicationInstanceIDs
<TOPIC_NAME>__DXPublicationInstanceDates
<TOPIC_NAME>__DXPublicationInstancePartitionDate
```

Where `<TOPIC_NAME>` is the name of the topic from which the subscriber consumes the data.

Step 5. Add a Filter Query to the Reader Object

If you are creating a mapping for an unbound subscription do not add a filter query to the mapping.

1. Configure a Filter query on the source with the following mapping parameters:

   ```
   <TOPIC_NAME>__DXPublicationInstanceIDs
   <TOPIC_NAME>__DXPublicationInstanceDates
   <TOPIC_NAME>__DXPublicationInstancePartitionDate
   ```

   Where `<TOPIC_NAME>` is the name of the topic from which the subscriber consumes the data.
   Do not enclose filter query parameters within quotation marks.

   For example, use the following format for a filter query with a filter condition on the partition date parameter:

   ```
   dih__publication_instance_id in ($MY_TOPIC__PublicationInstanceIDs) and
   dih__publication_instance_date in ($MY_TOPIC__PublicationInstanceDates) and
   publication_date_partition in ($MY_TOPIC__PublicationInstancePartition_Date)
   ```
2. Save the mapping.

Step 6. Configure the Mapping Run-time Environment and Create an Application

1. In the **Properties** pane select **Run-time**, and then, under **Validation Environments**, select **Hadoop** and then select **Hive**. Verify that **Native** is not selected.
2. Create an application from the mapping.

   The mapping is deployed to the Data Integration Service for Hadoop environment.

Developing Data Engineering Integration Workflows for Publications

To develop a Data Engineering Integration workflow for a publication, perform the following steps in the Developer tool:

1. Develop a Data Engineering Integration mapping for publications.
2. Create a workflow for publication and include the publication mapping or mappings that belong to the same topic. You can include multiple mappings in a single pipeline that is characterized by a Start_Event and End_Event.
3. Validate the mapping and workflow and deploy the workflow to an application in the Data Integration Service.
4. Create Data Engineering Integration workflows for publication in the Data Integration Hub console.

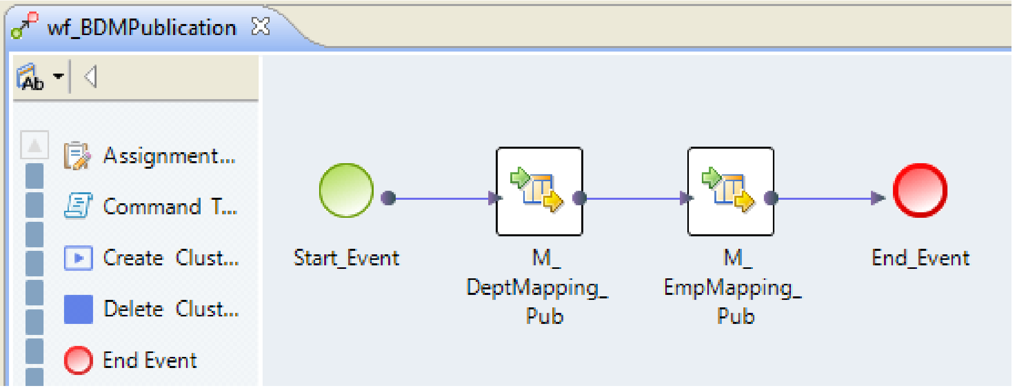
Step 1. Develop Data Engineering Integration Mappings for Publications

Develop Data Engineering Integration mappings for publications as described in "Developing Data Engineering Integration Mappings for Publications" on page 39.

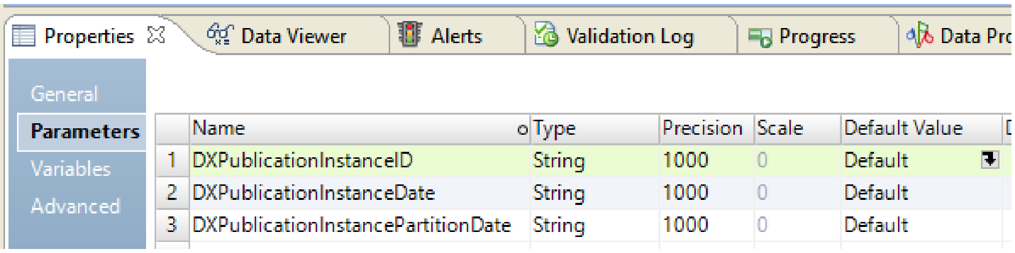
Step 2. Create a Publication Workflow

Create a publication workflow as follows:

1. Create a workflow for publication. You can include multiple mappings in a single pipeline within a Start_Event and an End_Event.

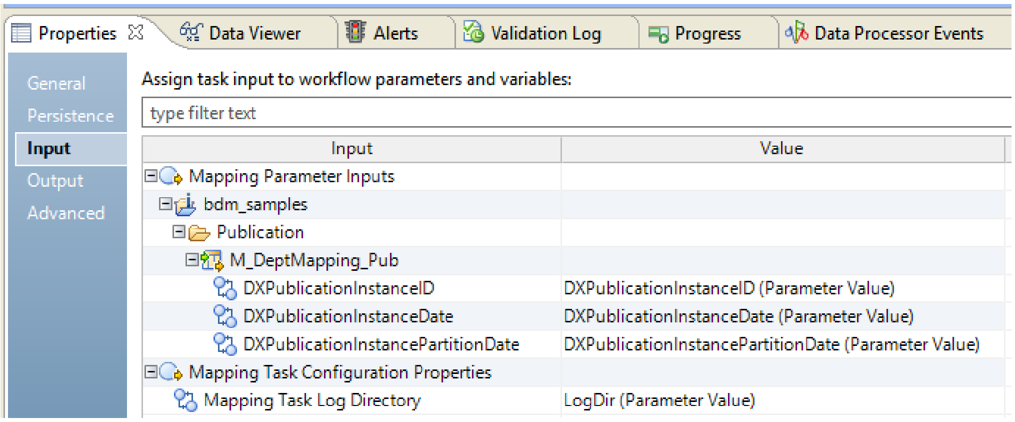


2. Create the following workflow parameters in the **Properties** > **Parameters** menu:
   - DXPublicationInstanceID
   - DXPublicationInstanceDate
   - DXPublicationInstancePartitionDate



	Name	Type	Precision	Scale	Default Value	D
1	DXPublicationInstanceID	String	1000	0	Default	
2	DXPublicationInstanceDate	String	1000	0	Default	
3	DXPublicationInstancePartitionDate	String	1000	0	Default	

3. Assign workflow parameters to corresponding mapping parameters for every mapping included within the workflow. Use the following workflow menu: **Workflow** > **Mapping** > **Properties** > **Input** > **Mapping Parameters**.



Assign task input to workflow parameters and variables:

Input	Value
Mapping Parameter Inputs	
bdm_samples	
Publication	
M_DeptMapping_Pub	
DXPublicationInstanceID	DXPublicationInstanceID (Parameter Value)
DXPublicationInstanceDate	DXPublicationInstanceDate (Parameter Value)
DXPublicationInstancePartitionDate	DXPublicationInstancePartitionDate (Parameter Value)
Mapping Task Configuration Properties	
Mapping Task Log Directory	LogDir (Parameter Value)

4. Save the workflow.

Step 3. Deploy the Publication Workflow

Select publication workflows and deploy them to the Data Integration Service of the Data Integration Hub console. The Data Integration Hub administrator can now access the workflow to create a Data Engineering Integration workflow for publications.

Developing Data Engineering Integration Workflows for Subscriptions

To develop a Data Engineering Integration workflow for a subscription, perform the following steps in the Developer tool:

1. Develop Data Engineering Integration Mapping for subscriptions.

2. Create a workflow for subscription and include the subscription mapping or mappings that belong to the same topic. You can include multiple mappings in a single pipeline that is characterized by a Start_Event and End_Event.
3. Validate the mapping and workflow and deploy the workflow to an application in the Data Integration Service.
4. Create Data Engineering Integration workflows for subscription in the Data Integration Hub console.

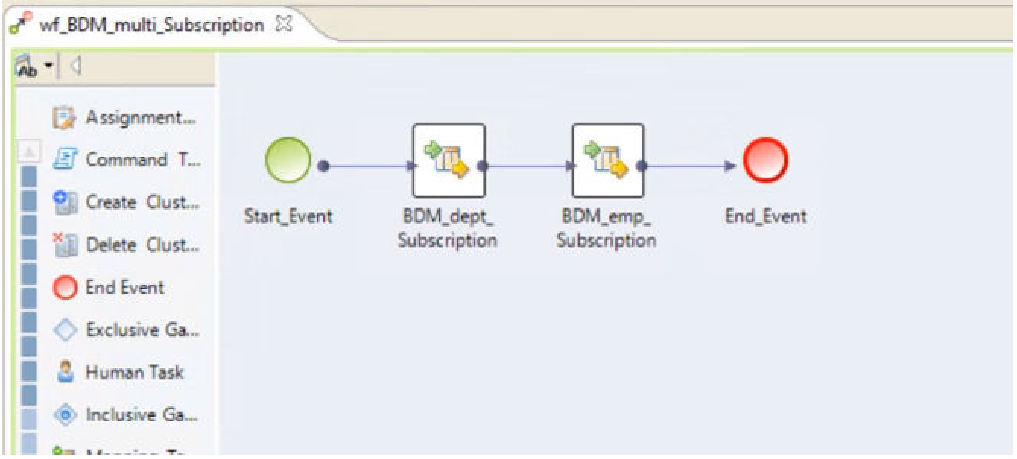
Step 1: Develop Data Engineering Integration Mappings for Subscriptions

Develop Data Engineering Integration mappings for subscriptions as described in "Developing Data Engineering Integration Mappings for Subscriptions" on page 43.

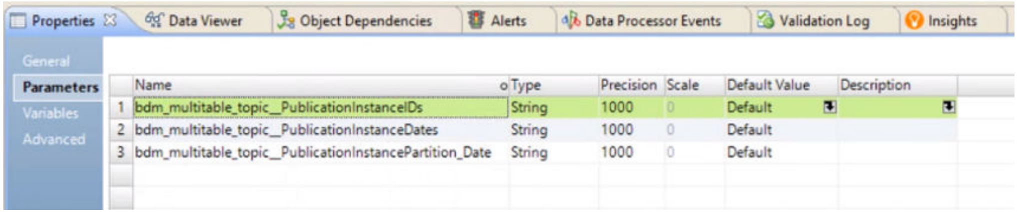
Step 2. Create a Subscription Workflow

Create a subscription workflow as follows:

1. Create a workflow for subscription. You can include multiple mappings in a single pipeline included within a Start_Event and an End_Event.

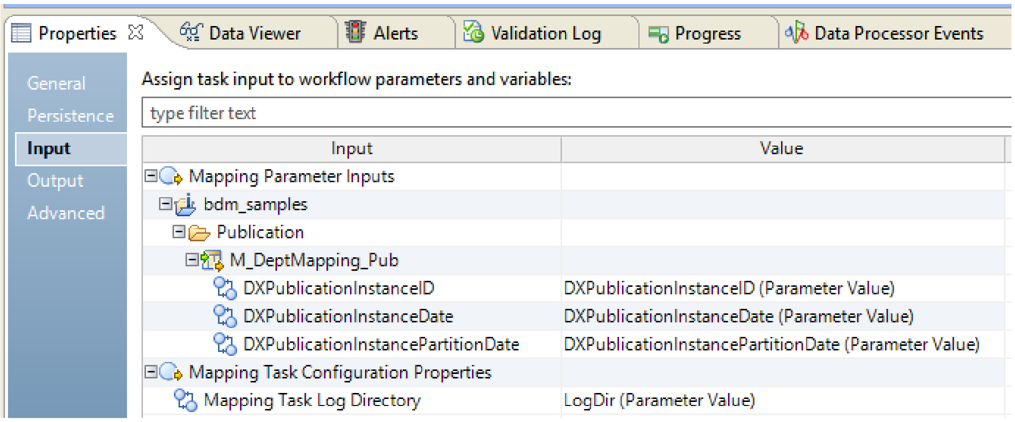


2. Create the following workflow parameters in the **Properties** > **Parameters** menu:
   - bdm_multiple_topic_PublicationInstanceIDs
   - bdm_multiple_topic_PublicationInstanceDates
   - bdm_multiple_topic_PublicationInstancePartition_Date



	Name	Type	Precision	Scale	Default Value	Description
1	bdm_multitable_topic__PublicationInstanceIDs	String	1000	0	Default	
2	bdm_multitable_topic__PublicationInstanceDates	String	1000	0	Default	
3	bdm_multitable_topic__PublicationInstancePartition_Date	String	1000	0	Default	

3. Assign workflow parameters to corresponding mapping parameters for every mapping included within the workflow. Use the following workflow menu: **Workflow** > **Mapping** > **Properties** > **Input** > **Mapping Parameters**.



Assign task input to workflow parameters and variables:

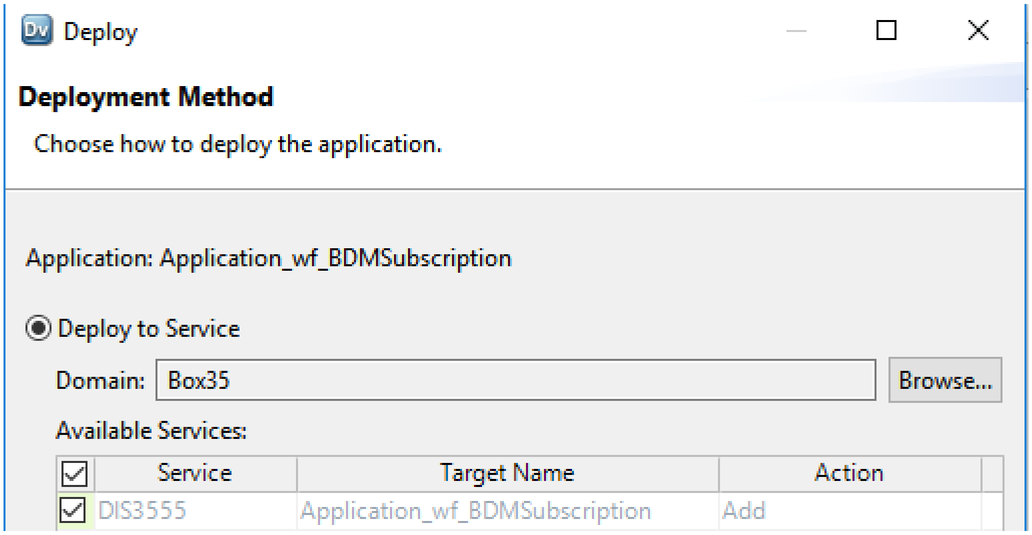
type filter text

Input	Value
Mapping Parameter Inputs	
bdm_samples	
Publication	
M_DeptMapping_Pub	
DXPublicationInstanceID	DXPublicationInstanceID (Parameter Value)
DXPublicationInstanceDate	DXPublicationInstanceDate (Parameter Value)
DXPublicationInstancePartitionDate	DXPublicationInstancePartitionDate (Parameter Value)
Mapping Task Configuration Properties	
Mapping Task Log Directory	LogDir (Parameter Value)

4. Save the workflow.

Step 3. Deploy the Subscription Workflow

Select subscription workflows and deploy them to the Data Integration Service of the Data Integration Hub console. The Data Integration Hub administrator can now access the workflow to create Data Engineering Integration workflows for subscriptions.



Deployment Method

Choose how to deploy the application.

Application: Application_wf_BDMSubscription

● Deploy to Service

Domain: Box35

Available Services:

☑	Service	Target Name	Action
☑	DIS3555	Application_wf_BDMSubscription	Add

CHAPTER 4

Data Quality Mappings and Workflows

This chapter includes the following topics:

Data Quality Mappings and Workflows Overview

Data Quality is a processing engine that Data Integration Hub uses to run Data Integration Hub custom publications and subscriptions for on-premise applications. The Data Integration Service runs the Data Quality mappings and workflows on the native environment.

You use the Developer tool to develop the Data Quality mappings that process the publications and subscriptions. You then use the Data Integration Hub Operation Console to import the Data Quality mapping into a Data Integration Hub workflow. For details, see "Creating a Data Integration Hub Workflow" on page 62.

You create a Data Quality workflow by using multiple Data Quality mappings.

The Data Integration Hub operator creates a publication or a subscription in the Data Integration Hub operation console, and selects the Data Integration Hub workflow which is based on the Data Quality mapping or Data Quality workflow. For more information, see the *Data Integration Hub Operator Guide*.

You can find sample mappings in the following directory: `<DIHInstallationDir>/samples/idq_mappings`. Each sample mapping has an associated readme file that describes the sample mapping and contains instructions.

Data Quality Mappings and Workflows in Data Integration Hub

You use Data Quality mappings and workflows in Data Integration Hub publications and subscriptions with custom mappings.

In publication mappings, the source is the publishing application and the target is the Data Integration Hub publication repository. In subscription mappings, the source is the Data Integration Hub publication repository and the target is the subscribing application.

You can use user-defined session parameters in custom workflows and define their values in Data Integration Hub or in a parameter file. You can manage the values of user-defined session parameters in Data Integration Hub in the Forms Designer. You cannot manage the values of built-in session parameters in Data Integration Hub. For more information about session parameters, see the section "Working with Session Parameters" in the *PowerCenter Advanced Workflow Guide*.

Note: To use Data Quality mappings and workflows to publish files with file transfer, session parameter names must start with $InputFile_.

Before You Begin

Before you develop Data Quality mappings and Data Quality workflows to use in Data Integration Hub custom big data publications and subscriptions, verify that the following conditions exist:

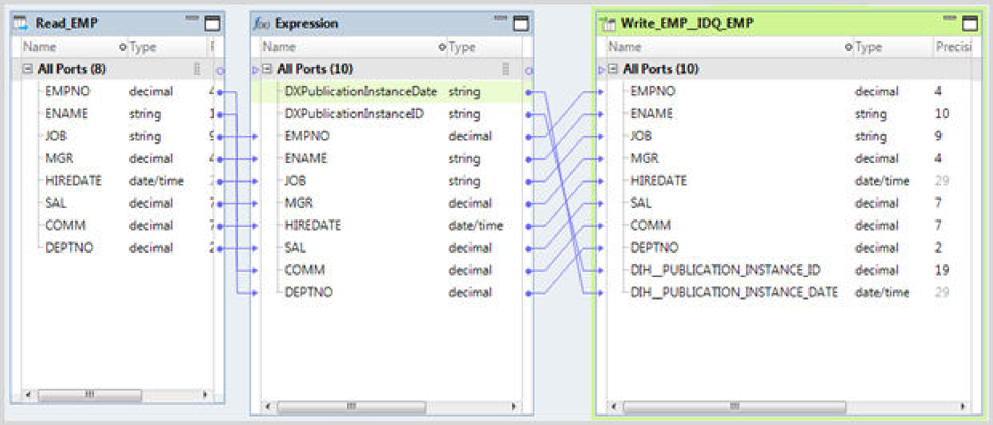
- The Informatica platform is installed in your organization.
- The Data Integration Hub Data Quality component is installed.
- The topic to which to publish data and from which to consume data is configured in Data Integration Hub.
- The following connections are configured in Data Integration Hub:
  - Connection to the publishing application.
  - Connection to the subscribing application.

Developing Data Quality Mappings for Publications

To develop a Data Quality mapping for a publication, perform the following steps in the Developer tool:

1. Create source and target connections. The source connection is a connection to the publishing application and the target connection is a connection to the Data Integration Hub publication repository.
2. Create source and target data objects.
3. Create a mapping and add the source and target objects to the mapping.
4. Add Data Integration Hub parameters to the mapping.
5. Add an Expression transformation to the mapping, configure ports in the transformation, and connect ports between the source, the transformation, and the target.
6. Configure the mapping run-time environment and create an application from the mapping.

The following image shows a sample publication mapping:



Read_EMP

Name	Type
All Ports (8)	
EMPNO	decimal
ENAME	string
JOB	string
MGR	decimal
HIREDATE	date/time
SAL	decimal
COMM	decimal
DEPTNO	decimal

Expression

Name	Type
All Ports (10)	
DXPublicationInstanceDate	string
DXPublicationInstanceID	string
EMPNO	decimal
ENAME	string
JOB	string
MGR	decimal
HIREDATE	date/time
SAL	decimal
COMM	decimal
DEPTNO	decimal

Write_EMP_IDQ_EMP

Name	Type	Precisi
All Ports (10)		
EMPNO	decimal	4
ENAME	string	10
JOB	string	9
MGR	decimal	4
HIREDATE	date/time	29
SAL	decimal	7
COMM	decimal	7
DEPTNO	decimal	2
DIH__PUBLICATION_INSTANCE_ID	decimal	19
DIH__PUBLICATION_INSTANCE_DATE	date/time	29

Step 1. Create Source and Target Connections

1. Create an Oracle or a Flat File source connection to the publishing application.
2. Create a relational target connection to the Data Integration Hub publication repository.

Step 2. Create Source and Target Data Objects

Create data objects under **Physical Data Objects**.

1. Create a source data object and select the table in the source connection to publish. The object can be a relational or a flat file data object.
2. Create a target data object and select the table in the target connection to where to publish the data from the source. The object must be a relational data object.

Step 3. Create a Mapping with Source and Target

1. Create and name a new mapping.
2. Add the source physical data object to the mapping as a Reader.
3. Add the target physical data object to the mapping as a Writer.

Step 4. Add Data Integration Hub Parameters to the Mapping

▶ Add the following parameters to the mapping:

```
DXPublicationInstanceID
DXPublicationInstanceDate
```

Step 5. Add an Expression Transformation to the Mapping

1. Add an Expression transformation to the mapping, between the source and target objects.
2. Link all the ports from the source object to the identical ports in the Expression transformation. For example, if your topic table includes the column `EMPNO`, link the `EMPNO` port in the source object to the `EMPNO` port in the Expression transformation.

3. Configure the following additional ports in the Expression transformation:

   ```
   DXPublicationInstanceID
   DXPublicationInstanceDate
   ```

   The data type of all ports is string, and the minimum precision is 200.

4. Link all the ports from the Expression transformation to the identical ports in the target object. For example, link the `EMPNO` port in the Expression transformation to the `EMPNO` port in the target object, and link the `DXPublicationInstanceID` port in the Expression transformation to the `dih__PublicationInstanceID` port in the target object.

5. Save the mapping.

Step 6. Configure the Mapping Run-time Environment and Create an Application

1. In the **Properties** pane select **Run-time**, and then, under **Validation Environments**, select **Native**.

2. Create an application from the mapping.

   The mapping is deployed to the Data Integration Service native environment.

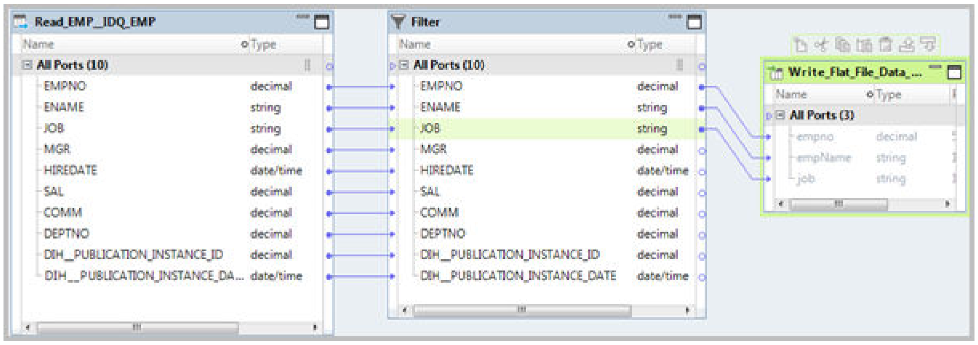
Developing Data Quality Mappings for Subscriptions

To develop a Data Quality mapping for a subscription, perform the following steps in the Developer tool:

1. Create source and target connections. The source connection is a connection to the Data Integration Hub publication repository and the target connection is a connection to the subscribing application.
2. Create source and target data objects.
3. Create a mapping and add the source and target objects to the mapping.
4. Add a Data Integration Hub parameter to the mapping.
5. Add a Filter transformation to the mapping, configure the transformation filter, and connect ports between the source, the transformation, and the target.
6. Add a Filter query to the Reader object.
7. Configure the mapping run-time environment and create an application from the mapping.

Note: If you are creating a mapping for an unbound subscription you do not need add a Filter transformation and a Filter query to the mapping.

The following image shows a sample subscription mapping:



Read_EMP__IDQ_EMP

Name	Type
All Ports (10)	
EMPNO	decimal
ENAME	string
JOB	string
MGR	decimal
HIREDATE	date/time
SAL	decimal
COMM	decimal
DEPTNO	decimal
DIH__PUBLICATION_INSTANCE_ID	decimal
DIH__PUBLICATION_INSTANCE_DA...	date/time

Filter

Name	Type
All Ports (10)	
EMPNO	decimal
ENAME	string
JOB	string
MGR	decimal
HIREDATE	date/time
SAL	decimal
COMM	decimal
DEPTNO	decimal
DIH__PUBLICATION_INSTANCE_ID	decimal
DIH__PUBLICATION_INSTANCE_DATE	date/time

Write_Flat_File_Data_...

Name	Type
All Ports (3)	
empno	decimal
empName	string
job	string

Step 1. Create Source and Target Connections

1. Create a relational source connection to the Data Integration Hub publication repository.
2. Create an Oracle or a Flat File target connection to the subscribing application.

Step 2. Create Source and Target Data Objects

Create data objects under **Physical Data Objects**.

1. Create a source data object and select the table in the source connection to consume. The object must be a relational data object.
2. Create a target data object and select the table in the target connection subscribes to the data. The object can be a relational data object or a flat file data object.

Step 3. Create a Mapping with Source and Target

1. Create and name a new mapping.
2. Add the source physical data object to the mapping as a Reader.
3. Add the target physical data object to the mapping as a Writer.

Step 4. Add Data Integration Hub Parameter to the Mapping

▶ Add the following parameter to the mapping:

```
<TOPIC_NAME>__DXPublicationInstanceIDs
```

Where `<TOPIC_NAME>` is the name of the topic from which the subscriber consumes the data.

Step 5. Add a Filter Transformation to the Mapping

Add a filter condition on the publication date. If you are creating a mapping for an unbound subscription do not add a Filter transformation to the mapping.

1. Add a Filter transformation to the mapping, between the source and target objects.
2. Link all the ports from the source object to the identical ports in the Filter transformation. For example, link the `EMPNO` port in the source object to the `EMPNO` port in the Filter transformation, and link the `DXPublicationInstanceID` port in the source object to the `DXPublicationInstanceID` port in the Filter transformation.
3. In the Filter tab of the Filter transformation, configure the following parameter:

   ```
   <TOPIC_NAME>__PublicationInstanceDates
   ```

   Where <TOPIC_NAME> is the name of the topic from which the subscriber consumes the data.

   Verify that **Default Value** is set to TRUE.
4. Save the mapping.

Step 6. Add a Filter Query to the Reader Object

Add a filter condition on the publication ID parameter. If you are creating a mapping for an unbound subscription do not add a filter query to the mapping.

1. Configure a Filter query on the source with the following mapping parameter:

   ```
   <TOPIC_NAME>__DXPublicationInstanceIDs
   ```

   Where `<TOPIC_NAME>` is the name of the topic from which the subscriber consumes the data. Do not enclose filter the query parameter within quotation marks.

   For example:

   ```
   dih__publication_instance_id in ($MY_TOPIC__PublicationInstanceIDs)
   ```

2. Save the mapping.

Step 7. Configure the Mapping Run-time Environment and Create an Application

1. In the **Properties** pane select **Run-time**, and then, under **Validation Environments**, select **Native**.
2. Create an application from the mapping.

   The mapping is deployed to the Data Integration Service native environment.

Developing Data Quality Workflows for Publications

To develop a Data Quality workflow for a publication, perform the following steps in the developer tool:

1. Develop Data Quality mappings for publications.
2. Create a workflow for publication and include the publication mapping or mappings that belong to the same topic. You can include multiple mappings in a single pipeline that is characterized by a Start_Event and End_Event.
3. Validate the mapping and workflow and deploy the workflow to an application in the Data Integration service.
4. Create Data Quality workflows for publication in the Data Integration Hub console.

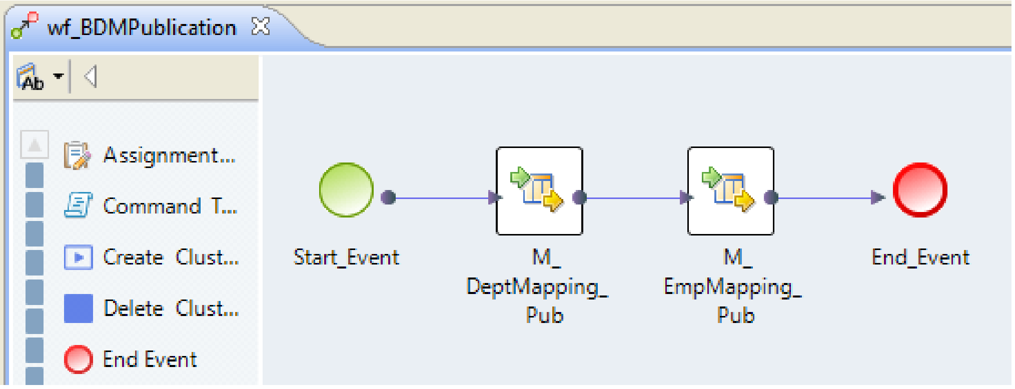
Step 1. Develop Data Quality Mapping for Publications

Develop Data Quality mapping for publications as described in "Developing Data Quality Mappings for Publications" on page 50.

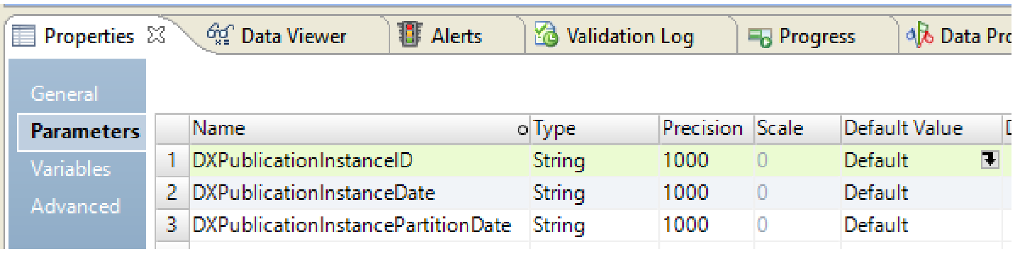
Step 2. Create a Publication Workflow

Create a publication workflow as follows:

1. Create a workflow for publication. You can include multiple mappings in a single pipeline within a Start_Event and an End_Event.

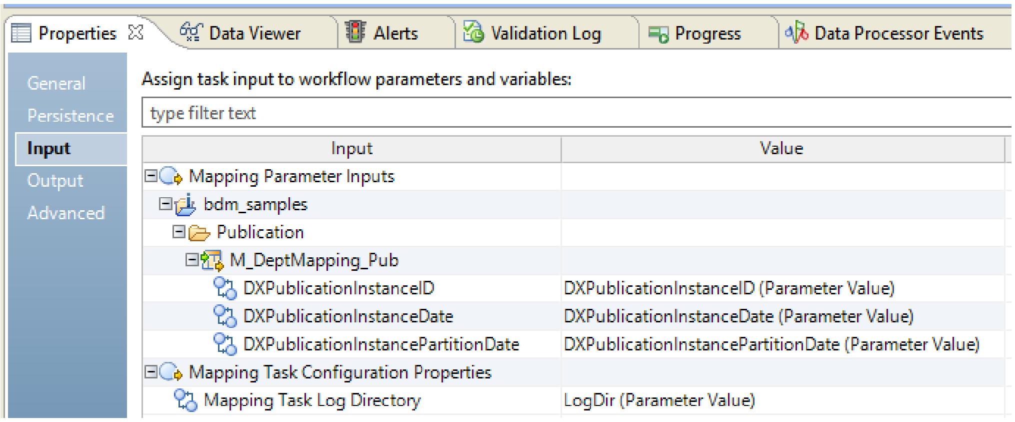


2. Create the following workflow parameters in the **Properties** > **Parameters** menu:
   - DXPublicationInstanceID
   - DXPublicationInstanceDate
   - DXPublicationInstancePartitionDate



	Name	Type	Precision	Scale	Default Value
1	DXPublicationInstanceID	String	1000	0	Default
2	DXPublicationInstanceDate	String	1000	0	Default
3	DXPublicationInstancePartitionDate	String	1000	0	Default

3. Assign workflow parameters to corresponding mapping parameters for every mapping included within the workflow. Use the following workflow menu: **Workflow** > **Mapping** > **Properties** > **Input** > **Mapping Parameters**.



Assign task input to workflow parameters and variables:

Input	Value
⊟ Mapping Parameter Inputs	
⊟ bdm_samples	
⊟ Publication	
⊟ M_DeptMapping_Pub	
DXPublicationInstanceID	DXPublicationInstanceID (Parameter Value)
DXPublicationInstanceDate	DXPublicationInstanceDate (Parameter Value)
DXPublicationInstancePartitionDate	DXPublicationInstancePartitionDate (Parameter Value)
⊟ Mapping Task Configuration Properties	
Mapping Task Log Directory	LogDir (Parameter Value)

4. Save the workflow.

Step 3. Deploy the Publication Workflow

▶ Select publication workflows and deploy them to Data Integration Service for the Data Integration Hub console. Data Integration Hub administrator can now access the workflow to create Data Quality workflow for publications.

Developing Data Quality Workflows for Subscriptions

To develop a Data Quality workflow for a subscriptions, perform the following steps in the developer tool:

1. Develop Data Quality mappings for subscriptions.
2. Create a workflow for subscription and include the subscription mapping or mappings that belong to the same topic. You can include multiple mappings in a single pipeline that is characterized by a Start_Event and End_Event.
3. Validate the mapping and workflow and deploy the workflow to an application in the Data Integration service.
4. Create Data Quality workflows for subscription in the Data Integration Hub console.

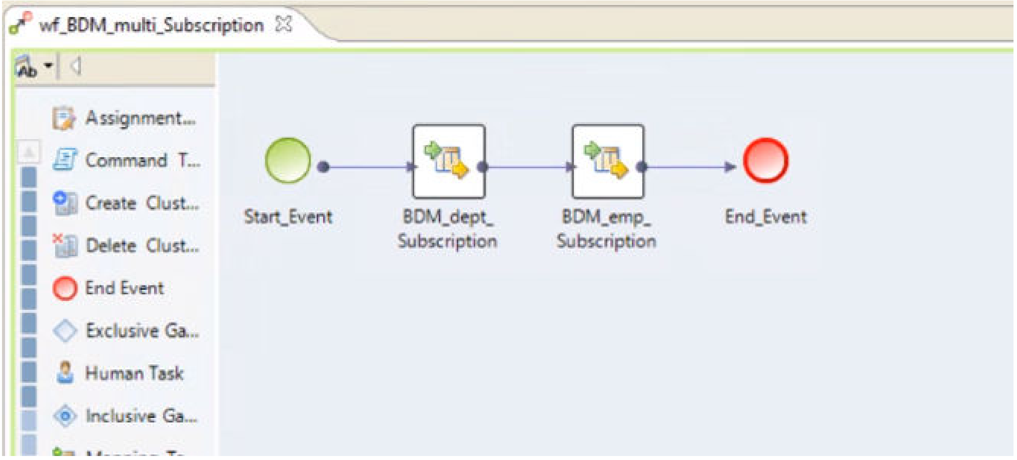
Step 1. Develop Data Quality Mapping for Subscriptions

Develop Data Quality mapping for subscriptions as described in "Developing Data Quality Mappings for Subscriptions" on page 52.

Step 2. Create a Subscription Workflow

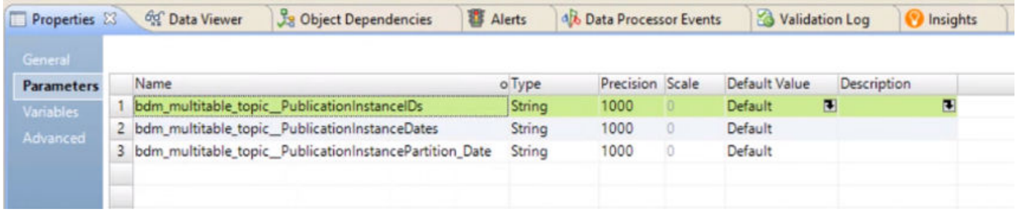
Create a subscription workflow as follows:

1. Create a workflow for subscription. You can include multiple mappings in a single pipeline included within a Start_Event and an End_Event.



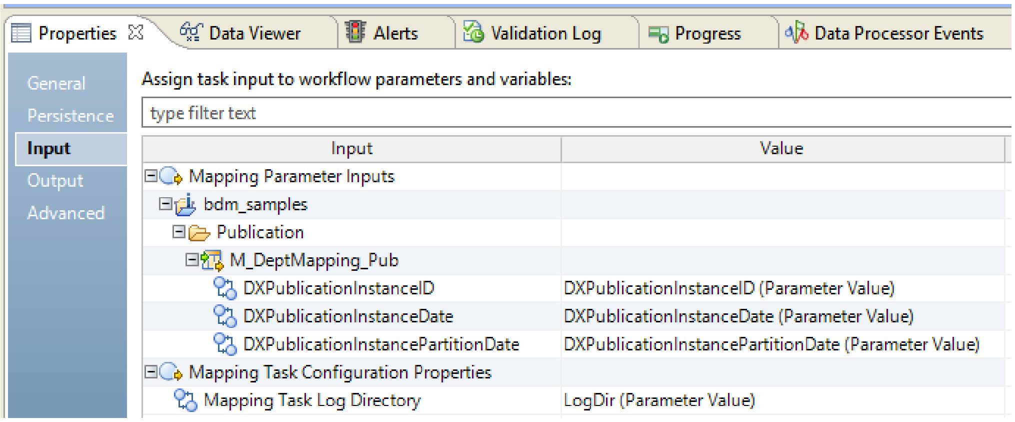
2. Create the following workflow parameters in the **Properties** > **Parameters** menu:
   - bdm_multiple_topic_PublicationInstanceIDs
   - bdm_multiple_topic_PublicationInstanceDates

- bdm_multiple_topic_PublicationInstancePartition_Date



	Name	Type	Precision	Scale	Default Value	Description
1	bdm_multitable_topic__PublicationInstanceIDs	String	1000	0	Default	
2	bdm_multitable_topic__PublicationInstanceDates	String	1000	0	Default	
3	bdm_multitable_topic__PublicationInstancePartition_Date	String	1000	0	Default	

3. Assign workflow parameters to corresponding mapping parameters for every mapping included within the workflow. Use the following workflow menu: **Workflow** > **Mapping** > **Properties** > **Input** > **Mapping Parameters**.



Assign task input to workflow parameters and variables:

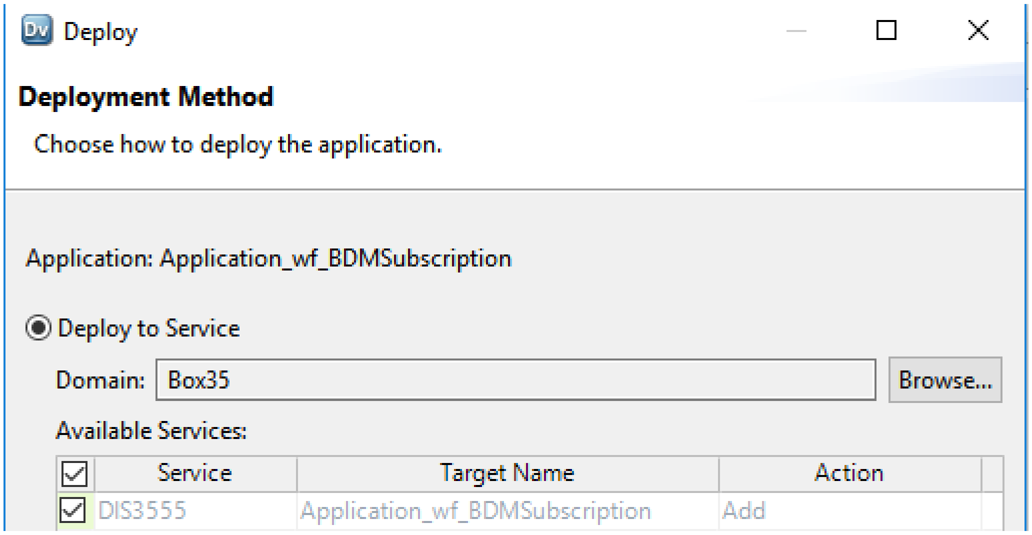
Input	Value
Mapping Parameter Inputs	
bdm_samples	
Publication	
M_DeptMapping_Pub	
DXPublicationInstanceID	DXPublicationInstanceID (Parameter Value)
DXPublicationInstanceDate	DXPublicationInstanceDate (Parameter Value)
DXPublicationInstancePartitionDate	DXPublicationInstancePartitionDate (Parameter Value)
Mapping Task Configuration Properties	
Mapping Task Log Directory	LogDir (Parameter Value)

4. Save the workflow.

Step 3. Deploy the Subscription Workflow

▶ Select subscription workflows and deploy them to the Data Integration Service for the Data Integration Hub console.

The Data Integration Hub administrator can now access the workflow to create a Data Quality workflow for subscriptions.



Deployment Method

Choose how to deploy the application.

Application: Application_wf_BDMSubscription

● Deploy to Service

Domain: Box35

Available Services:

☑	Service	Target Name	Action
☑	DIS3555	Application_wf_BDMSubscription	Add

CHAPTER 5

Informatica Cloud Mappings and Tasks

This chapter includes the following topics:

Informatica Cloud Mappings and Tasks Overview

Data Integration Hub uses Informatica Cloud mappings and tasks to publish data from source cloud applications to the Data Integration Hub publication repository and to consume data from the publication repository by target cloud applications.

In publications and subscriptions with automatic mappings, you use Informatica Cloud mappings to define the data processing.

In publications and subscriptions with automatic mappings, you use Data Synchronization tasks and Mapping Configuration tasks to define the data processing.

You develop Informatica Cloud mappings and tasks for Data Integration Hub in the same way that you develop other Informatica Cloud mappings and tasks. You use the Data Integration Hub connection as the target in publication mappings and tasks and as the source in subscription mappings and tasks.

Before you develop mappings and tasks in Informatica Cloud for Data Integration Hub, verify that the cloud Data Integration Hub connector is installed and a connection that uses the connector exists. For details, see the *Data Integration Hub Administrator Guide*.

Informatica Cloud Mappings in Data Integration Hub

You use Informatica Cloud mappings in Data Integration Hub publications and subscriptions with automatic mappings.

In publication mappings, the publishing application is the source of the mapping and the Data Integration Hub publication repository is the target of the mapping. In subscription mappings, the Data Integration Hub publication repository is the source of the mapping and the subscribing application is the target of the mapping.

You can create a generic, parameterized mapping and use it for multiple publications and subscriptions. You can parameterize any element of the mapping, for example, source and target connections, field mappings, and transformations. The operator fills in the parameter values when they create the publication or subscription in the Data Integration Hub Operation Console.

Informatica Cloud Mappings Rules and Guidelines

When you develop Informatica Cloud mappings to use in Data Integration Hub publications and subscriptions with automatic mappings, consider the following rules and guidelines:

- Do not run mappings that you create for Data Integration Hub from within Informatica Cloud. You must run Data Integration Hub mappings from Data Integration Hub by running the publication or the subscription to which the mapping is associated.
- When you use the Data Integration Hub connection, if the mapping is not parametized, the target object in a publication mapping and the source object in a subscription mapping presents the list of topics defined in Data Integration Hub. The format of the list is `TopicName.tableName`.
- If a mapping is parameterized, you can use it as both a publication mapping and a subscription mapping.
- If a mapping is not parameterized, and you want to distinguish between publication mappings and subscription mappings, indicate the type of the mapping in the mapping name. For example, name a publication mapping Pub_<MappingName>, and name a subscription mapping Sub_<MappingName>.
- You can add expression and filter transformations to the mapping.
- If the mapping includes parameters, when the operator creates the publication or subscription, the parameters show in the **Input Parameters** tab of the publication or subscription wizard.

Data Integration Tasks in Data Integration Hub

You use Data Integration tasks in Data Integration Hub publications and subscriptions with custom mappings.

In publication tasks, the publishing application is the source of the task and the Data Integration Hub publication repository is the target of the task. In subscription tasks, the Data Integration Hub publication repository is the source of the task and the subscribing application is the target of the task.

When you develop a Data Integration task for a Data Integration Hub publication, you select the Data Integration Hub connection as the target object. When you develop a Data Integration task for a Data Integration Hub subscription, you select the Data Integration Hub connection as the source object.

Note: You do not use Data Integration tasks in data-driven publications and subscriptions.

Data Integration Tasks Rules and Guidelines

When you develop Data Integration mappings and tasks to use in Data Integration Hub publications and subscriptions with custom mappings, consider these rules and guidelines.

General rules and guidelines

Consider the following rules and guidelines when you create Data Integration mappings and tasks:

- Do not run tasks that you create for Data Integration Hub from within Informatica Intelligent Cloud Services. You must run the tasks from Data Integration Hub by running the publication or the subscription to which the task is associated.
- When you use the Data Integration Hub connection, the target object in a publication mapping or task and the source object in a subscription mapping or task presents the list of topics defined in Data Integration Hub. The format of the list is `TopicName.tableName`.
- Data Integration Hub determines the scheduling of the publication or the subscription based on the settings that the operator defined for the publication or the subscription. When you create the Data Integration task, in the **Schedule** page of the task wizard, verify that the option **Do not run this task on a schedule** is selected.
- To distinguish between publication tasks and subscription tasks, indicate the type of the task in the task name. When you select a task for a publication or for a subscription, you can easily select an appropriate task.
  For example, name a publication task Pub_<TaskName>, and name a subscription task Sub_<TaskName>.

Synchronization task rules and guidelines

Consider the following rules and guidelines when you create synchronization tasks and mappings:

- The task operation for publication tasks is an insert operation.
- When you create a publication task, select the Data Integration Hub connection as the target. When you create a subscription task, select the Data Integration Hub connection as the source.
- Synchronization tasks do not support multiple sources. Therefore, when you create a synchronization task for a publication or a subscription with multiple sources, create a relationship between the sources for the following use cases:
  - Publications: when you publish from multiple tables.
  - Subscriptions: when you subscribe to multiple tables, or when the subscription is a compound subscription.

Mapping task rules and guidelines

Consider the following rules and guidelines when you create mapping tasks and mappings:

- The mapping operation is an insert operation for both publication and subscription mappings.
- When you create a publication mapping, select the Data Integration Hub connection when you configure the target properties. When you create a subscription mapping, select the Data Integration Hub connection when you configure the source properties.

For instructions on creating Informatica Intelligent Cloud Services mappings and tasks for Data Integration Hub, see the *Data Integration Hub Connector Guide*.

Using Intelligent Structure Model in Data Integration Hub

Data Integration Hub supports unstructured flat files to publish data. You can use intelligent structure models to structure data from unstructured or semi-structured files and create custom cloud mappings that use intelligent structure models and publish unstructured data to Data Integration Hub.

For more information about how to use intelligent structure model in Data Integration Hub to publish unstructured or semi-structured data refer to the H2L ...

CHAPTER 6

Data Integration Hub Workflows

This chapter includes the following topics:

Data Integration Hub Workflows Overview

A Data Integration Hub workflow is a set of instructions that define one of the following actions:

- How to process data that on-premise applications publish to and consume from the Data Integration Hub publication repository with custom publications and subscriptions. A Data Integration Hub workflow for custom publications and subscriptions can be based on a PowerCenter workflow, Data Engineering Integration mapping and workflows, Data Engineering Streaming mapping, or Data Quality mapping and workflows.
- How to pre-process data before you publish it. A Data Integration Hub pre-processing workflow must be based on a PowerCenter workflow.
- How to post-process data to which you subscribe. A Data Integration Hub post-processing workflow must be based on a PowerCenter workflow.
- What action to take when publication or subscription events are in a defined status, as defined in event monitoring rules that invoke a PowerCenter workflow. A Data Integration Hub workflow for monitoring rules must be based on a PowerCenter workflow.

Before you create a workflow in Data Integration Hub, you must create the relevant PowerCenter workflow, Data Engineering Integration mapping or workflow, Data Engineering Integration streaming, Data Quality mapping or workflow. For more information, see the following sections:

- Chapter 2, “PowerCenter Mappings and Workflows” on page 22
- Chapter 3, “Data Engineering Integration and Streaming Mapping and Workflows” on page 37
- Chapter 4, “Data Quality Mappings and Workflows” on page 49

You then select the workflow or mapping when you create the Data Integration Hub workflow. For more information, see “Creating a Data Integration Hub Workflow” on page 62.

When the operator creates the publication or subscription or the monitoring rule, they select the Data Integration Hub workflows to use, as applicable.

You do not need to create a Data Integration Hub workflow for the following publication and subscription types:

- Publications and subscriptions that use an automatic mapping. Data Integration Hub creates a PowerCenter workflow and mapping directly from the publication or subscription.
- Publications and subscriptions for cloud applications. The operator selects an Informatica Intelligent Cloud Services task when they create the publication or the subscription.

Workflow Permissions

Workflow permissions control who can access the workflow.

The Data Integration Hub administrator creates categories and assigns categories to user groups to determine the users that have permissions to view, edit, and delete workflows. When you create or edit a workflow, you can select categories for the workflow. If you select categories for a workflow, only users that belong to user groups that have at least one of the categories that you select have permissions for the workflow. If you don't select categories for a workflow, all users have permissions for the workflow.

Workflow Management

Use the **Create Workflow** page to create, edit, and delete Data Integration Hub workflows.

Data Integration Hub operators use the workflows with custom publications and subscriptions, to pre-process data before you publish it or post-process data to which you subscribe, and to define the action to take when publication or subscription events are in a defined status.

Creating a Data Integration Hub Workflow

Use the Navigator to create a Data Integration Hub workflow.

After you create the workflow, the Data Integration Hub can associate the workflow with a publication or a subscription in the publication or subscription wizard, or associate the workflow with a monitoring rule in the monitoring rule wizard, as applicable.

To create a Data Integration Hub workflow, perform the following tasks:

1. Define general workflow parameters.
2. If the workflow contains user-defined parameters, optionally, customize the parameters.
3. Optionally, add event attributes to the workflow.
4. Optionally, define workflow permissions.

Step 1. Define General Workflow Properties

Define general workflow properties on the **General** tab of the **Create Workflow** page.

1. On the Navigator, click **Hub Management** > **Workflows** > **New Workflow**.
2. Enter the workflow name. The name can contain up to 64 characters and can include spaces and special characters.

3. Optionally, enter a description of the workflow.
4. Select the type of processor to run the workflow. You can select one of the following processor types:
    - Select **Data Engineering Integration Mapping** to associate the Data Integration Hub workflow with a Data Engineering Integration mapping. The mapping runs once and stops after completion. You can assign Data Engineering Integration mappings to custom batch publications that publish big data from on-premises applications to Hadoop-based topics, and to custom batch subscriptions that consume big data from Hadoop-based topics into on-premises applications.
    - Select **Data Engineering Integration Workflow** to associate the Data Integration Hub workflow with a Data Engineering Integration workflow. You can assign Data Engineering Integration workflows to custom batch publications that publish big data from on-premises applications to Hadoop-based topics, and to custom batch subscriptions that consume big data from Hadoop-based topics into on-premises applications.

      Note: Data Engineering Integration Workflow supports publication and subscription usage only.
    - Select **Data Engineering Streaming Real-time Mapping** to associate the Data Integration Hub workflow with a Data Engineering Streaming mapping. The mapping runs continuously and reads from streaming sources. You can assign Data Engineering Streaming mappings to custom multi-latency publications that publish big data streams from on-premises applications to Hadoop-based topics.
    - Select **Data Quality Mapping** to associate the Data Integration Hub workflow with a Data Engineering Quality mapping. The mapping runs once and stops after completion. You can assign Data Engineering Quality mappings to custom publications that publish from on-premises applications to database- or file-based topics and to custom subscriptions that consume data from database- or file-based topics into on-premises applications.
    - Select **Data Quality Workflow** to associate the Data Integration Hub workflow with a Data Engineering Quality workflow. The workflow runs once and stops after completion. You can assign Data Engineering Quality workflow to custom publications that publish from on-premises applications to database- or file-based topics and to custom subscriptions that consume data from database- or file-based topics into on-premises applications.

      Note: Data Engineering Quality Workflow supports publication and subscription usage only.
    - Select **PowerCenter Batch Workflow** to associate the Data Integration Hub workflow with a PowerCenter batch workflow. A PowerCenter batch workflow reads from a file, database, or another source. The workflow runs once and stops after completion. You can assign PowerCenter batch workflows to custom publications that publish from on-premises applications to database- or file-based topics, to custom subscriptions that consume data from database- or file-based topics into on-premises applications, to publication pre-processes, to subscription post-processes, and to event monitoring rules that invoke a PowerCenter workflow.
    - Select **PowerCenter Real-time Workflow** to associate the Data Integration Hub workflow with a real-time workflow. A PowerCenter real-time workflow runs continuously and reads from real-time sources. You can assign PowerCenter real-time workflows to custom publications that publish from on-premises applications to database- or file-based topics. 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.
5. Select the type of process to use the Data Integration Hub workflow for. Select one of the following options:
    - Publication Pre-Processing. Runs a publication pre-process.

- Publication. Runs a publication process.
- Subscription. Runs a subscription process.
- Subscription Post-Processing. Runs a subscription post-process.
- Monitoring rule. Runs the workflow when publication or subscription events are in a defined status, as defined in the event monitoring rule.

6. Select the workflow or mapping to associate with the Data Integration Hub workflow. Your selection depends on the type of workflow or mapping that you select in **Processor Type**:
   - When the mapping type is **Data Engineering Integration mapping**, the **Add a Data Engineering Integration Mapping** dialog box shows. To show all the Data Engineering Integration mappings that are deployed in the Data Integration Service for Hadoop environment, click **Show All**. To search for a mapping by application name, enter a string in the **Application name** text box and then click **Search**. Select a mapping and then click **Add**.
   - When the mapping type is **Data Engineering Streaming Real-time mapping**, the **Add a Data Engineering Streaming Real-time Mapping** dialog box shows. To show all the Data Engineering Streaming mappings that are deployed in the Data Integration Service for Hadoop environment, click **Show All**. To search for a mapping by application name, enter a string in the **Application name** text box and then click **Search**. Select a mapping and then click **Add**.
   - When the mapping type is **Data Engineering Quality mapping**, the **Add an Data Engineering Quality Mapping** dialog box shows. To show all mappings that are deployed in the Data Integration Service for native environment, click **Show All**. To search for a mapping by application name, enter a string in the **Application name** text box and then click **Search**. Select a mapping and then click **Add**.
   - When the mapping type is a PowerCenter workflow, either a batch workflow or a real-time workflow, the **Add a PowerCenter Workflow** dialog box shows. Select one of the following options:
     - **Select a workflow from the PowerCenter repository**. Select a PowerCenter workflow that you saved in the PowerCenter repository and then click **Add**. 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 and then click **Add**. To select a file, click the browse button, browse to the location of the required file, select the file, and then click **Upload**.
7. If you do not want to define workflow parameters, event attributes, or permissions, click **Save**.

Step 2. Customize Workflow Parameters

If the workflow contains parameters, you can add values to the parameters and customize the layout and behavior of the parameters with the Forms Designer. Access the Forms Designer on the **Workflow Parameters** tab of the **Create Workflow** page.

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.

1. Click the **Workflow Parameters** tab.

   If the workflow or mapping that is associated with the Data Integration Hub workflow contains user-defined parameters, the **Preview** pane displays the default view of the workflow parameter list.

   Note: Do not edit parameter values on the **Preview** pane.
2. Click **Designer**.

   The **Designer** pane displays the parameter list in alphabetic order.

3. To arrange the parameters and groups, drag the element or group you want to arrange.
4. To edit parameter values and to change the behavior and appearance of a parameter element, click the parameter and click **Actions** > **Edit**.

   The **Details** dialog box appears.
5. Change the properties for the parameter and click **Save**.
6. To add a group, click the **Actions** menu and choose to add a tab, column, header, or disclosure.
7. To preview the customized form, click **Preview** and verify the appearance of the form and any field dependencies you defined.
8. If you do not want to define event attributes or permissions for the workflow, click **Save**.

Step 3. Add Event Attributes to the Workflow

Add event attributes to the workflow on the **Event Attributes** tab of the **Create Workflow** page.

For more information about event attributes, see the *Data Integration Hub Administrator Guide*.

1. Click the **Event Attributes** tab.
2. In the **Available Attribute Keys** list select the attributes to export and click the Select icon. You can select a single attribute or multiple attributes.

   The attributes you selected appear in the **Selected Attribute Keys** list.
3. If you do not want to define permissions for the workflow, click **Save**.

Step 4. Define Workflow Permissions

Control access to the workflow on the **Permissions** tab of the **Create Workflow** page.

1. Click the **Permissions** tab.
2. In the **Available Categories** list select the categories to add to the workflow and click the Select icon. You can select a single category or multiple categories.

   The categories you selected appear in the **Selected Categories** list.
3. Click **Save**.

Editing a Data Integration Hub Workflow

Use the Navigator to edit a Data Integration Hub workflow.

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

   The **Workflows** page appears.
2. Click the Edit icon next to the workflow and update workflow properties as required.
3. Click **Save**.

Deleting a Data Integration Hub Workflow

Use the Navigator to delete a Data Integration Hub workflow.

When you delete the workflow, you also delete all of the dependent objects of the workflow.

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

   The **Workflows** page appears.

2. Click the Delete icon next to the workflow and confirm the deletion.

Data Integration Hub Workflow Properties

The **Create Workflow** page contains the following tabs:

General tab

Define basic workflow properties and select the processor type, workflow usage, and mapping.

Workflow Parameters tab

If the workflow contains user-defined parameters, you can add values to the parameters and customize them.

Event Attributes tab

Associate event attributes with the workflow.

Permissions tab

Control access to the workflow in the Operation Console by selecting categories to grant users permissions to access the workflow. Only users that belong to user groups that have at least one of the categories that you select have permissions for the workflow. If you do not select any category, all users have permissions for the workflow.

Workflow General Properties

The **General** tab of the **Create Workflow** page includes the following properties:

Workflow Name

Name of the workflow. The name can contain up to 64 characters and can include spaces and special characters.

Description

Optional description of the workflow.

Processor Type

Consider the following guidelines when you select the processor type:

- Select **Data Engineering Integration Mapping** to associate the Data Integration Hub workflow with a Data Engineering Integration mapping. The mapping runs once and stops after completion. You can assign Data Engineering Integration mappings to custom batch publications that publish big data from on-premises applications to Hadoop-based topics, and to custom batch subscriptions that consume big data from Hadoop-based topics into on-premises applications.
- Select **Data Engineering Integration Workflow** to associate the Data Integration Hub workflow with a Data Engineering Integration workflow. You can assign Data Engineering Integration workflows to custom batch publications that publish big data from on-premises applications to Hadoop-based topics, and to custom batch subscriptions that consume big data from Hadoop-based topics into on-premises applications.

  Note: Data Engineering Integration Workflow supports publication and subscription usage only.
- Select **Data Engineering Streaming Real-time Mapping** to associate the Data Integration Hub workflow with a Data Engineering Streaming mapping. The mapping runs continuously and reads

from streaming sources. You can assign Data Engineering Streaming mappings to custom multi-latency publications that publish big data streams from on-premises applications to Hadoop-based topics.

- Select **Data Quality Mapping** to associate the Data Integration Hub workflow with a Data Engineering Quality mapping. The mapping runs once and stops after completion. You can assign Data Engineering Quality mappings to custom publications that publish from on-premises applications to database- or file-based topics and to custom subscriptions that consume data from database- or file-based topics into on-premises applications.
- Select **Data Quality Workflow** to associate the Data Integration Hub workflow with a Data Engineering Quality workflow. The workflow runs once and stops after completion. You can assign Data Engineering Quality workflow to custom publications that publish from on-premises applications to database- or file-based topics and to custom subscriptions that consume data from database- or file-based topics into on-premises applications.

  Note: Data Engineering Quality Workflow supports publication and subscription usage only.
- Select **PowerCenter Batch Workflow** to associate the Data Integration Hub workflow with a PowerCenter batch workflow. A PowerCenter batch workflow reads from a file, database, or another source. The workflow runs once and stops after completion. You can assign PowerCenter batch workflows to custom publications that publish from on-premises applications to database- or file-based topics, to custom subscriptions that consume data from database- or file-based topics into on-premises applications, to publication pre-processes, to subscription post-processes, and to event monitoring rules that invoke a PowerCenter workflow.
- Select **PowerCenter Real-time Workflow** to associate the Data Integration Hub workflow with a real-time workflow. A PowerCenter real-time workflow runs continuously and reads from real-time sources. You can assign PowerCenter real-time workflows to custom publications that publish from on-premises applications to database- or file-based topics. 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.

Usage

Type of process to use the Data Integration Hub workflow for. Select one of the following options:

- Publication Pre-Processing. Runs a publication pre-process.
- Publication. Runs a publication process.
- Subscription. Runs a subscription process.
- Subscription Post-Processing. Runs a subscription post-process.
- Monitoring rule. Runs the workflow when publication or subscription events are in a defined status, as defined in the event monitoring rule.

Mapping

The workflow or mapping to add to Data Integration Hub. The dialog box that shows depends on the type of workflow or mapping that you select in **Processor Type**:

- When the mapping type is **Data Engineering Integration mapping**, the **Add a Data Engineering Integration Mapping** dialog box shows. To show all the Data Engineering Integration mappings that are deployed in the Data Integration Service for Hadoop environment, click **Show All**. To search for a mapping by application name, enter a string in the **Application name** text box and then click **Search**. Select a mapping and then click **Add**.

- When the mapping type is **Data Engineering Streaming Real-time mapping**, the **Add a Data Engineering Streaming Real-time Mapping** dialog box shows. To show all the Data Engineering Streaming mappings that are deployed in the Data Integration Service for Hadoop environment, click **Show All**. To search for a mapping by application name, enter a string in the **Application name** text box and then click **Search**. Select a mapping and then click **Add**.
- When the mapping type is **Data Engineering Quality mapping**, the **Add an Data Engineering Quality Mapping** dialog box shows. To show all mappings that are deployed in the Data Integration Service for native environment, click **Show All**. To search for a mapping by application name, enter a string in the **Application name** text box and then click **Search**. Select a mapping and then click **Add**.
- When the mapping type is a PowerCenter workflow, either a batch workflow or a real-time workflow, the **Add a PowerCenter Workflow** dialog box shows. Select one of the following options:
  - **Select a workflow from the PowerCenter repository**. Select a PowerCenter workflow that you saved in the PowerCenter repository and then click **Add**. 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 and then click **Add**. To select a file, click the browse button, browse to the location of the required file, select the file, and then click **Upload**.

Workflow Parameters Properties

The **Workflow Parameters** tab of the **Create Workflow** page includes the following views:

Preview

A preview of workflow parameters.

Designer

You can customize the layout and behavior of the parameters with the 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.

Workflow Event Attributes Properties

The **Event Attributes** tab of the **Create Workflow** page includes the following properties:

Available Attribute Keys

List of attribute keys that you can assign to the workflow.

Selected Attribute Keys

List of attribute keys assigned to the workflow.

Workflow Permissions Properties

The **Permissions** tab of the **Create Workflow** page includes the following properties:

Available Categories

List of categories that you can assign to the workflow.

Selected Categories

List of categories assigned to the workflow.

CHAPTER 7

Data Integration Hub Transformations

This chapter includes the following topics:

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 Integration Hub 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.

When you use the transformation in a subscription post-processing workflow, you must notify the Data Integration Hub server that the subscription post-process is complete and send the post-processing status to the server. The Data Integration Hub server then creates a child event for the subscription that evoked the post-processing workflow, with the status of the post-process.

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 successfully. 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 64 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.
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 8

Operational Data Store Dashboard and Reports

This chapter includes the following topics:

Operational Data Store Dashboard and Reports Overview

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

You can make the Dashboard available in Data Integration Hub in two ways. When you install Data Integration Hub, you can enable the dashboard that uses operational data store. Otherwise, you can use the dashboard that uses metadata repository which is available by default. This section applies only if you use the Dashboard using operational data store.

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

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

Note: For general information about working with Logi Analytics, see Logi Analytics documentation.

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

The Data Integration Hub administrator installs the operational data store with the main Data Integration Hub installation and modifies Data Integration Hub system properties to determine certain aspects of the Dashboard behavior, such as whether to show the Dashboard when users log on to the Operation Console and the error rate range in the Error Rate Gauge panel. Operators can personalize the Dashboard page to display different panels and apply filters based on the information that the operators want to view.

After the Data Integration Hub administrator installs the Dashboard and Reports component, they use PowerCenter Workflow Manager to configure the operational data store event loader. The operational data store event loader is a PowerCenter workflow that collects KPIs from the Data Integration Hub repository according to specified parameters and loads aggregated events to the operational data store. In the operational data store event loader, the Data Integration Hub administrator defines parameters such as the number of retry attempts for each event load process in case of failure or the number of minutes to wait between event load processes.

Key Performance Indicators (KPIs)

This section applies only if you use the Dashboard using operational data store. Use key performance indicators (KPIs) to collect information about events that Data Integration Hub generates. The operational data store event loader collects KPIs from the main Data Integration Hub repository and transfers aggregated KPIs to the Dashboard.

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

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

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

Default KPIs

This section applies only if you use the Dashboard using operational data store. The operational data store uses default KPIs in reports. The operational data store event loader collects the KPIs and loads the KPIs to the operational data store.

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

The following table describes the default KPIs that the Dashboard using operational data store displays:

KPI	Description
Message processing time	Duration in minutes of the time it takes for the event to initially reach a final state.
Number of events	Number of events that Data Integration Hub processes.
Number of error events	Number of error events that reached a final state.

Dashboard and Reports Structure in Logi Info Studio

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

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

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

Operational Data Store Dashboard Filters

This section applies only if you use the Dashboard using operational data store. The Dashboard uses filters for each report that you generate in Logi Info Studio. The filters correspond the global filters in the Dahsboard page of the Operation Console.

The following table describes the default Dashboard filters:

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

Operational Data Store Dashboard SQL Query Elements

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

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

Note: You cannot create reports in Logi Info Studio with SQL queries to the run-time Data Integration Hub repository.

The following table describes the default Dashboard SQL constants:

SQL Constant Name	Description
DX_ODS_FILTER_ACCOUNT	Filters the database rows according to the related account. Includes the following query parameters: - requestAccountId
DX_ODS_FILTER_ERROR_STATE	Filters the database rows to display only error events.
DX_ODS_FILTER_EVENT_STATUS	Filters the database rows according to the event status. Includes the following query parameters: - requestEventStatusId
DX_ODS_FILTER_EVENT_STATUS_STATE	Filters the database rows according to the event state. The following values are available: - **1**. Show error events with a final state. - **2**. Show non-error events with a final state. - **-1**. Show all events with a final state. Includes the following query parameters: - requestEventStatusState
DX_ODS_FILTER_EVENT_TYPE	Filters the database rows according to the event type. Includes the following query parameters: - requestEventTypeId
DX_ODS_FILTER_PARTNER	Filters the database rows according to the related partner. Includes the following query parameters: - requestPartnerId
DX_ODS_FILTER_TIME_ORACLE	Filters the database rows according to a selected time frame. Includes the following query parameters: - requestDateBegin - requestTimeBegin - requestDateEnd - requestTimeEnd

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

Operational Data Store Dashboard JavaScript Support Files

This section applies only if you use the Dashboard using operational data store. Data Integration Hub uses JavaScript files and jQuery files to process event information when you view the Dashboard in the Operation Console.

The files are located in the following location:

```
<DIHInstallationDir>\DataIntegrationHub\tomcat\webapps\dih-dashboard\_SupportFiles
```

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

File	Description
dx.js	Functions that the Data Integration Hub server uses when it processes reports with the LogiXML engine.
dxbrowser.js	Functions that the browser client uses when you view the Dashboard in the Operation Console.
jquery-*.*	Collection of jQuery files that enable dynamically-populating drop down lists in the global filter section, such as partner or account.

Operational Data Store Dashboard Theme Modifier

This section applies only if you use the Dashboard using operational data store The theme modifier controls the appearance of the Dashboard panels. Data Integration Hub applies the style and appearance settings in the theme modifier to any custom Dashboard panel that you create.

The theme modifier files are located in the following location:

```
<DIHInstallationDir>\DataIntegrationHub\tomcat\webapps\dih-dashboard\_Themes\DIHDashboard
\Informatica
```

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

File	Description
Theme.css	Cascading style sheet that determines the graphical and textual formatting of the Dashboard panels.
ThemeModifier.xml	Global settings for the Dashboard panels behavior and the single sign-on mechanism that Data Integration Hub uses when you log on to the Operation Console.

Custom Dashboard Settings

This section applies only if you use the Dashboard using operational data store. You modify certain properties in the LogiXML _Settings.lgx file when you create a custom Dashboard for Data Integration Hub.

The following table describes the properties to modify:

Property	Description
DX_CONSOLE_URL	Location of the Operation Console, in the following format: `http://<DIH host>:<DIH port>/DIH-console`
DASHBOARD_SAVEFOLDER	Location to store the personalized layout settings for each Dashboard user. Default is: `@Function.AppPhysicalPath~\dx\saved_dashboards`
DX_ODS_ORACLE_CONNECTION	Connection string for the operational data store.
DX_ORACLE_CONNECTION	Connection string for the main Data Integration Hub repository.
LogonFailPage	URL to display after the user logs out of the Operation Console. The default URL is the Data Integration Hub Logout page: `http://<DIH host>:<DIH port>/DIH-console/logout.jsp`
AuthenticationClientAddresses	IP addresses for servers that can request authentication tokens from the LogiXML server, separated by semicolons. For each server, set IP4 and IP6 addresses.
caption	Description of the custom Dashboard.
COOKIE_PATH	Path to the cookie file for the custom Dashboard. Must match the value in the **Cookie Path** property.

Property	Description
WEBAPP_NAME	Name of the folder which contains the custom Dashboard. Default is: **dxdashboard**
Redirect Error URL	Link for the page to display if the Dashboard encounters an error.

Custom Dashboard and Reports in Logi Info Studio

This section applies only if you use the Dashboard using operational data store. Use Logi Info Studio to create custom Dashboard panels based on default KPIs or custom KPIs.

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

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

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

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

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

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

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

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

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

Installing and Configuring Logi Info Studio

This section applies only if you use the Dashboard using operational data store. You install Logi Info Studio to enhance and create Dashboard panels for Data Integration Hub.

Before you install Logi Info Studio, verify that the Data Integration Hub Dashboard and Reports component is installed on the Data Integration Hub machine and that your system meets the minimum requirements for Logi Info Studio. For more information about the system requirements for Logi Info Studio, refer to Logi Analytics documentation.

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

Setting Up the Custom Dashboard

This section applies only if you use the Dashboard using operational data store. You create the custom Dashboard as an application in Logi Info Studio and import the default Dashboard files to Logi Info Studio.

1. In Logi Info Studio, create a Java application.
2. In the **Prepare New Application** wizard, configure the name and location for the Dashboard application:
   - In the application name, enter the name for the custom Dashboard.
   - In the deployment folder location, enter the following path:

     ```
     <DIHInstallationDir>\DataIntegrationHub\tomcat\webapps
     ```
3. Skip the remaining **Prepare a New Application** wizard screens.
4. Copy the contents of the `<DIHInstallationDir>\DataIntegrationHub\tomcat\webapps\DIH-dashboard` folder to the newly created `<DIHInstallationDir>\DataIntegrationHub\tomcat\webapps\<custom_dashboard>` folder.
5. Copy the LogiXML license file from the `<DIHInstallationDir>\DataIntegrationHub\tomcat\webapps\DIH-dashboard` folder to the `<DIHInstallationDir>\DataIntegrationHub\tomcat\webapps\<custom_dashboard>` folder.
6. In the LogiXML _Settings.lgx file, modify the properties according to the name and location of the custom Dashboard.
7. In the Data Integration Hub server, replace the value of the dx.dashboard.url system property with the value of the URL for the custom dashboard in the following format: `http://<host>:<port>/<custom_dashboard>`

Creating a Report in the Custom Dashboard Application

This section applies only if you use the Dashboard using operational data store. After you set up the custom Dashboard in Logi Info Studio, you create and configure a report to use in a custom Dashboard panel. You can copy an existing report and modify it as needed.

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

Adding a Dashboard Panel to the Custom Dashboard

This section applies only if you use the Dashboard using operational data store. After you create a report in Logi Info Studio, you add the custom Dashboard panel and its elements to the main Dashboard report file.

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

Deploying and Testing the Custom Dashboard in B2B Data Exchange

This section applies only if you use the Dashboard using operational data store. After you modify or create a custom Dashboard in Logi Info Studio, you copy the custom Dashboard application folder contents to Data Integration Hub. You then test that the custom Dashboard appears correctly.

1. Copy the entire custom Dashboard application folder to the following location: `<DIHInstallationDir>\DataIntegrationHub\tomcat\webapps`.
2. Back up the _Setting.lgx file and modify the following settings:
   - Delete the value of the **Redirect Error URL** property to allow access to the LogiXML error report.
   - Set the value of the **Debugger Style** property to **DebuggerLinks** to add links to the LogiXML progress report. The Dashboard panel layout may display differently.
3. Log on to the Operation Console and test that the new Dashboard panels open and populate correctly.
4. After you test the custom Dashboard, restore the properties in the _Setting.lgx file.

CHAPTER 9

Forms Designer

This chapter includes the following topics:

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.

CHAPTER 10

Data Integration Hub Publications and Subscriptions APIs

This chapter includes the following topics:

Publications and Subscriptions APIs Overview

Use the Data Integration Hub publications and subscriptions APIs to run publications and subscriptions, to enable and disable publications and subscriptions, to reprocess publication and subscription events, to query the status of publication and subscription events, and to extract data from the Data Integration Hub catalog.

You can use the following publications and subscriptions APIs:

- Run Publication Subscription REST API. Starts a publication or a subscription, including disabled publications and subscriptions, and returns the event ID of the publication or the subscription event that Data Integration Hub generates.

  Note: You can use the Run Publication Subscription REST API to publish data and subscribe to data with publications and subscriptions that trigger a Data Integration task. You cannot use the API to publish data and subscribe to data with data-driven publications and subscriptions.

- Run Publication Subscription command line API. Starts a publication or a subscription from a command line utility and returns the event ID of the publication or the subscription event that Data Integration Hub generates. You cannot run disabled publications and subscriptions with the command line API. You cannot call the command line API from a remote server.

  Tip: Use the Run Publication Subscription REST API to start the running of publications and subscriptions from a remote server.

- Change Publication Subscription Mode REST API. Changes the mode of a publication or a subscription, that is, enables a disabled publication or subscription and disables an enabled publication or subscription.

- Reprocess Event REST API. Reprocesses a publication or subscription event, including events of disabled publications and subscriptions.
- Event Status REST API. Returns the status of a publication or subscription event.

Run Publication Subscription REST API

Use the Data Integration Hub Run Publication Subscription REST API to run a specific publication or a specific subscription. You can run the publication or the subscription regardless of its mode, that is, you can run both enabled and disabled publications and subscriptions.

Note: You can use the Run Publication Subscription REST API to publish data and subscribe to data with publications and subscriptions that trigger a Data Integration task. You cannot use the API to publish data and subscribe to data with data-driven publications and subscriptions.

The Run Publication Subscription API returns the response code of the action that you perform. If the publication or subscription runs successfully, the API returns 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 run a publication or a subscription from the API, 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*.

Sample Files

The `<DIH_InstallationDir>\samples\rest_api_samples` folder contains sample client applications and a readme file that describes the applications.

Run Publication Subscription REST API Request

Data Integration Hub uses different REST URLs for running a publication and for running a subscription.

To run a publication, use the following REST URL:

```
http://<DIH_console_host_port>/dih-console/api/v1/publication/start
```

For example:

```
http://localhost:18080/dih-console/api/v1/publication/start
```

To run a subscription, use the following REST URL:

```
http://<DIH_console_host_port>/dih-console/api/v1/subscription/start
```

For example:

```
http://localhost:18080/dih-console/api/v1/subscription/start
```

Note: The start subscription REST API does not support concurrent triggers of the same subscription. It is recommended that you trigger the subscription after a couple of seconds.

Request syntax for running a publication

To run a publication, use the following request syntax:

```
{
            "publicationName": "<publicationName>",
            "parameters": [
              {"name": "param1", "value": "value"},
              {"name": "param2", "value": "value"}
            ],
            "paramFileContent": "<content of parameter file>",
            "runPreprocess": "<true/false>",
            "runDisabled": "<true/false>"
}
```

The following list describes the elements of the request:

- `publicationName`. Name of the publication to run.
- `parameters`. Optional request parameters.
- `paramFileContent`. Content of optional parameter file.
- `runPreprocess`. Whether or not to run a publication pre-process before the publication runs.
- `runDisabled`. Whether or not to run a publication that is in a Disabled status.

For example:

```
{
            "publicationName": "daily_sales",
            "parameters": [
              {"name": "SRC_SAP", "value": "sapbw.etl-tools.info"},
              {"name": "TGT_ORA", "value": "oracle_sap_staging"}
            ],
            "paramFileContent": "[DWH_PROJECT.WF:wkf_daily_loading] \n $
$platform=rh7",
            "runPreprocess": "false",
            "runDisabled": "true"
}
```

Request syntax for running a subscription

To run a subscription, use the following request syntax:

```
{
            "subscriptionName": "<subscriptionName>",
            "parameters": [
              {"name": "param1", "value": "value"},
              {"name": "param2", "value": "value"}
            ],
            "paramFileContent": "<content of parameter file>",
            "runDisabled": "<true/false>"
}
```

The following list describes the elements of the request:

- `subscriptionName`. Name of the subscription to run.
- `parameters`. Optional request parameters.
- `paramFileContent`. Content of optional parameter file.
- `runDisabled`. Whether or not to run a subscription that is in a Disabled status.

For example:

```
{
            "subscriptionName": "daily_report",
            "parameters": [
              {"name": "SRC_ORA", "value": "oracle_sap_staging"},
              {"name": "SuccessEmail", "value": "customer@informatica.com"}
            ],
            "paramFileContent": "[DWH_PROJECT.WF:wkf_daily_loading] \n $
$platform=rh7",
            "runDisabled": "true"
}
```

Run Publication Subscription REST API Action Response

When you use the Data Integration Hub Run Publication Subscription REST API to start the running of a publication or of a subscription, Data Integration Hub returns the response code of the action that you perform in the REST API response.

Running a publication or a subscription from the REST API returns one of the following response codes:

- 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.
- FAILED. Data Integration Hub could not trigger the publication or the subscription. The response provides the reason for the failure. For example, Data Integration Hub did not run the subscription because no publications are ready for consumption by the subscription.

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.

Tip: Use the Run Publication Subscription REST API to start the running of publications and subscriptions from a remote server.

The Run Publication Subscription API returns the response code of the action that you perform. If the publication or subscription runs successfully, the API returns 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 run a publication or a subscription from the API, 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 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.
-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.

Option	Argument	Description
-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.

Change Publication Subscription Mode REST API

Use the Data Integration Hub Change Publication Subscription Mode REST API to change the mode of a publication or a subscription, that is, to enable a disabled publication or subscription or to disable an enabled publication or subscription.

To change the mode of a publication, use the following REST URL:

```
http://<DIH_console_host_port>/dih-console/api/v1/publication/changemode
```

For example:

```
http://localhost:18080/dih-console/api/v1/publication/changemode
```

To change the mode of a subscription, use the following REST URL:

```
http://<DIH_console_host_port>/dih-console/api/v1/subscription/changemode
```

For example:

```
http://localhost:18080/dih-console/api/v1/subscription/changemode
```

Request syntax for changing the mode of a publication

To change the mode of a publication, use the following request syntax:

```
{
            "publicationName": "<publicationName>",
            "mode": "<enable/disable>"
}
```

For example:

```
{
            "publicationName": "daily_sales",
            "mode": "enable"
}
```

Request syntax for changing the mode of a subscription

To change the mode of a subscription, use the following request syntax:

```
{
            "subscriptionName": "<subscriptionName>",
            "mode": "<enable/paused/disable>"
}
```

For example:

```
{
            "subscriptionName": "daily_reports",
            "mode": "disable"
}
```

Sample File

The `<DIH_InstallationDir>\samples\rest_api_samples` folder contains a sample client application and a readme file that describes the application.

Change Publication Subscription Mode REST API Action Response

When you use the Data Integration Hub REST API to change the mode of a publication or of a subscription, Data Integration Hub returns the response code of the action that you perform in the REST API response.

Changing the mode of a publication or a subscription from the REST API returns one of the following response codes:

- When Data Integration Hub changes the mode of the publication or the subscription successfully, the API returns a SUCCESS response.
- When Data Integration Hub fails to change the mode of the publication or the subscription, the response provides the reason for the failure. For example, when you do not have sufficient privileges to perform the operation.

Reprocess Event REST API

Use the Data Integration Hub Reprocess Event REST API to reprocess publication and subscription events, including events of disabled publications and subscriptions.

To reprocess a publication or a subscription event, use the following REST URL:

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

For example:

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

Use the following syntax to reprocess an event:

```
{
            "eventId" : "<eventId>"
}
```

For example:

```
{
            "eventId" : "40558"
}
```

Sample File

The `<DIH_InstallationDir>\samples\rest_api_samples` folder contains a sample client application and a readme file that describes the application.

Reprocess Event REST API Action Response

When you use the Data Integration Hub Reprocess Event REST API to reprocess a publication or a subscription event, Data Integration Hub returns the response code of the action that you perform in the REST API response.

The response includes the following information:

Property	Description
responseCode	Response of the action: - `0`: success. - Any number higher than 0: error.
reprocessEventId	New event ID that Data Integration Hub generates for the publication or for the subscription when it reprocesses the existing event.
message	Error message. If the response code is `0` (success), the API returns the message `null`.

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 REST API, Data Integration Hub returns the event ID in the REST API response.
- 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 REST 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 a GET command with 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.
- `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.

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

Property	Description
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.
sourceSuccessRows	Number of source rows that Data Integration Hub read successfully.
sourceFailedRows	Number of source rows that Data Integration Hub failed to read.
targetFailedRows	Number of target rows that Data Integration Hub failed to write.
targetSuccessRows	Number of target rows that Data Integration Hub wrote successfully.
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,
    "sourceSuccessRows": 10,
    "sourceFailedRows": 0,
    "targetFailedRows": 0,
    "targetSuccessRows: 10}
```

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,
    "sourceSuccessRows": 15,
    "sourceFailedRows": 0,
    "targetFailedRows": 0,
    "targetSuccessRows: 15
}
```

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,
    "sourceSuccessRows": 2,
    "sourceFailedRows": 1,
    "targetFailedRows": 1,
    "targetSuccessRows: 2
    "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."
}
```

Change Event Status API

Use the Data Integration Hub Change Event Status API to change the status of an existing event.

To change the status of an event, use the following API URL:

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

For example:

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

You must provide the required header, body, and authorization details.

Request syntax for changing the status of an event

To change the status of an existing event, use the following request body syntax:

```
{
"eventId": "<event ID>",
"eventStatusName": "<name of the event status>",
"comment": "<Specify a comment to be displayed.>"
}
```

For example:

```
{
"eventId": "270003",
"eventStatusName": "Complete",
"comment": "From Rest API"
}
```

Sample File

The `<DIH_InstallationDir>\samples\rest_api_samples` folder contains a sample client application and a readme file that describes the application.

Change Event Status API Response

When you use the Data Integration Hub API to change the status of an existing event, Data Integration Hub returns the response code of the action that you perform in the API response.

Changing the event status using the REST API returns one of the following response codes:

- When Data Integration Hub changes the event status of an event successfully, the API returns a SUCCESS response.
  For example:

  ```
  SUCCESS
  {
  "responseCode": "SUCCESS"
  }
  ```

- When Data Integration Hub fails to change the event status of an event, the response provides the reason for the failure. For example, when you do not have sufficient privileges to perform the operation.
  For example:

  ```
  INSUFFICIENT_PRIVILEGES
  {
  "responseCode": "INSUFFICIENT_PRIVILEGES",
  "message": "The user does not have sufficient privileges or permissions to perform
  this operation on the event."
  }
  ```

CHAPTER 11

Data Extraction APIs

This chapter includes the following topics:

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.

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/dih-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 JSON string that contains metadata about all the topics for which you have the required privileges.

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 the topic, an entry with the table name and detailed information about each of the table fields.

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"
                    "tableFields":[
                        {
                            "name":"_Name_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                        },
                        {
                            "name":"_Type_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                        },
                        {
                            "name":"_ParentId_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                        }
                }
            ],
            "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"
                    "tableFields":[
                        {
                            "name":"_Name_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                        },
                        {
                            "name":"_Type_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                        },
                        {
                            "name":"_ParentId_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                           },
                        {
                            "name":"_StartDate_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                           },
                        {
                            "name":"_EndDate_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                        }
                },
                {
                    "tableName": "CustomerTable"
                    "tableFields":[
                        {
                            "name":"_Name_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                        },
                        {
                            "name":"_Type_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                        },
                        {
                            "name":"_ParentId_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                           },
                        {
                            "name":"_ExpectedRevenue_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                           },
                        {
                            "name":"_IsActive_"
                            "fieldType":"STRING"
                            "nullable":false
                            "scale":-1
                            "precision":255
                            "length":255
                            "primaryKey":false
                            "filterAccelerator":false
                            "encrypted":true
                        }
                },
                {

                },
            ],
            "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