Informatica

Informatica® Data Integration Hub
10.5

Administrator Guide

Informatica Data Integration Hub Administrator Guide
10.5
April 2021

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.

Publication Date: 2023-02-16

Table of Contents

Preface

Use the *Data Integration Hub Administrator Guide* to learn about the concepts, components, and tasks that you need to perform in the Data Integration Hub Operation Console. This guide includes information about the event types, user policies, and system settings. It also includes information on managing entities, repositories, dashboards, and reports.

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.

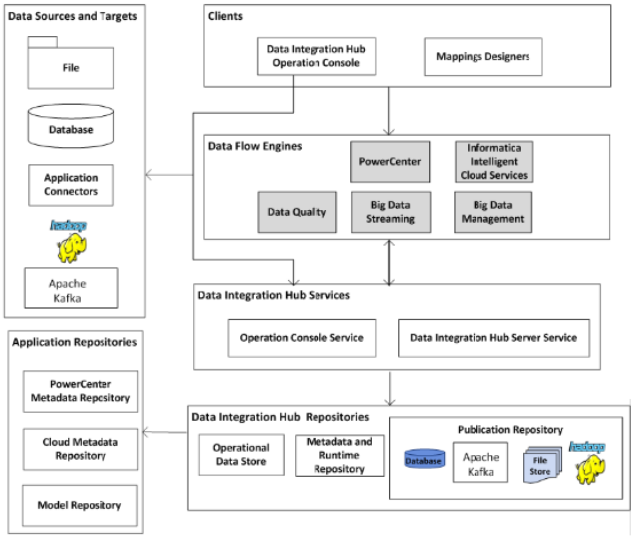
You can share sales data from Salesforce, marketing data from the marketing application, and user interaction data from the corporate website with the BI application as follows:

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.

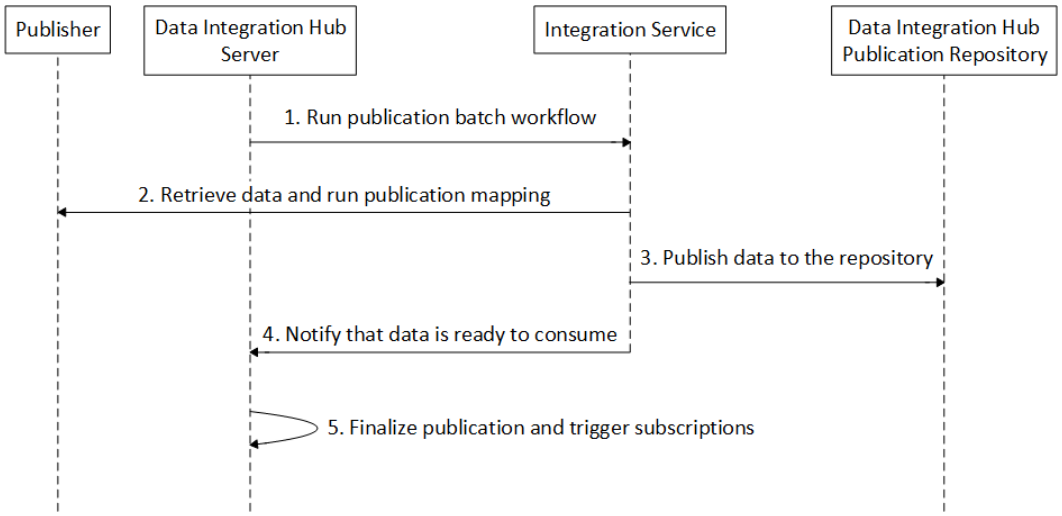
Publication Process with a Batch Workflow

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:



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.

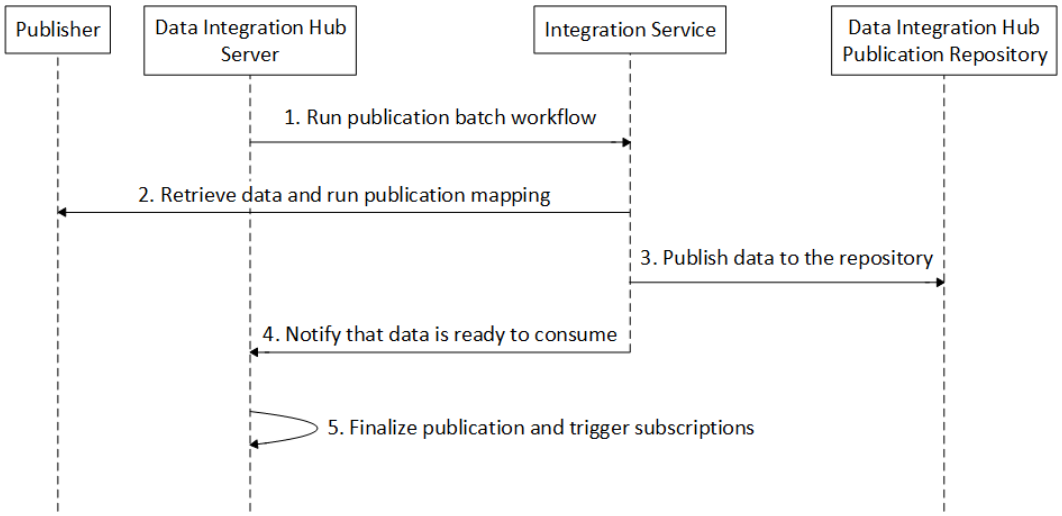
- Automatic subscriptions can run a Data Integration Hub workflow that is based on a PowerCenter batch workflow or run over a REST API.
- Custom subscriptions can either run a Data Integration Hub workflow that is based on a PowerCenter batch workflow, Data Engineering Integration mapping or workflow, 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:



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.

Administrator User Role

The Data Integration Hub administrator manages the administrative entities in the Operation Console. The entities include event types and statuses, user policies, and system settings.

The administrator creates and modifies custom event statuses, types, and attributes. The developer uses custom event types and event attributes when developing PowerCenter workflows for custom mappings. The operator uses custom event statuses in the Operation Console to process events.

The administrator configures and maintains user authentication with Informatica domain authentication or native authentication. The administrator also manages permissions and privileges that control the actions and objects that user groups can access in Data Integration Hub.

The administrator installs, configures, and maintains Data Integration Hub, including setting system properties, configuration properties, connections and ports, private keys, and Operation Console preferences.

To connect cloud-based applications to Data Integration Hub and use Informatica Intelligent Cloud Services for publications and subscriptions, the administrator installs and configures the Informatica Intelligent Cloud Services Data Integration Hub Connector, configures a connection to use in Informatica Intelligent Cloud Services tasks, and configures Data Integration Hub system properties for cloud connectivity.

The administrator creates, upgrades, and maintains the Data Integration Hub repository, Data Integration Hub publication repository, and the operational data store. The administrator also exports entities from and imports entities to the Data Integration Hub repository.

CHAPTER 2

Security

Security Overview

Data Integration Hub security protects the Data Integration Hub infrastructure against unauthorized access to or modifications of Data Integration Hub services and resources.

Infrastructure security includes the following aspects:

- Control Data Integration Hub server shutdown and startup by restricting remote shutdown and startup to specific clients in your organization.
- A secret token authorizes calls from PowerCenter workflows to the Data Integration Hub server through Data Integration Hub transformations. You can define a custom token to replace the default system token.
- Manage SSH File Transfer Protocol (SFTP) passwords and keys using a passphrase-protected keystore. Data Integration Hub uses Advanced Encryption Standard (AES) to encrypt the SSH private passwords and keys.

Restrict Remote Data Integration Hub Server Shutdown and Startup

You can define a list of clients from which users can remotely shut down and start up the Data Integration Hub server.

Users can still ping the Data Integration Hub server from any client in the network.

If you do not define the list, users can shut down and start up the server from any client in the network.

Shutdown and startup from the local host are enabled regardless of whether or not you define the list.

Configuring Clients for Remote Server Shutdown and Startup

1. On the Data Integration Hub server, open the following security configuration file:

   ```
   <DIHInstallationDir>\conf\security\dx-security-config.properties
   ```

2. In the security configuration file, in the `dx.security.dxcontrol.whitelist` property, enter the IP addresses of the clients from which users can remotely shut down and start up the Data Integration Hub server, separated by a semi colon.

   For example:

   ```
   dx.security.dxcontrol.whitelist=192.168.1.1;192.168.1.2;192.168.1.3;fe80::3516:cd0c:6
   f8:df39%19;
   ```

Define a Custom Secret Token for Calls to the Data Integration Hub Server

Customize the secret token that Data Integration Hub uses to authorize calls from PowerCenter workflows to the Data Integration Hub server through Data Integration Hub transformations.

The token you define replaces the default system token.

Defining a Custom Secret Token

1. In your PowerCenter Integration Service, create the following environment variable:

   ```
   dx.security.flowservice.shared_secret
   ```

   Assign the environment variable a value that will act as the shared secret between server and client.
2. On the Data Integration Hub server, open the following security configuration file:

   ```
   <DIHInstallationDir>\conf\security\dx-security-config.properties
   ```
3. In the security configuration file set the value of the property `dx.security.flowservice.shared_secret` to the shared secret that you assigned in step 1.

Secure Management of SFTP Passwords and Keys

Use the Data Integration Hub Security Keytool command line API to manage SSH File Transfer Protocol (SFTP) passwords and keys using a passphrase-protected keystore. Data Integration Hub uses Advanced Encryption Standard (AES) to encrypt the SSH private passwords and keys.

You can run the Security Keytool command line API from the Data Integration Hub server. To run the API you must provide Data Integration Hub Administrator user credentials.

You must restart the Data Integration Hub server and the Data Integration Hub Operation Console after you run the API. You cannot perform consecutive runs of the API without a server an Operation Console restart between the runs.

Data Integration Hub Security Keytool Command Line API Command Syntax

The Data Integration Hub Security Keytool Command Line API uses the following syntax:

```
dx-keytool
<-c|--command> rollKey|rollPassphrases|rollConfig|testConfig
<-u|--user> userID
```

```
<-p|--password> password
[--server "<hostname:port>"]
[-old_ksp|--oldKeystorePassphrase <current keystore passphrase>]
[-old_kp|--oldKeyPassphrase  <current key passphrase>]
[-ksp|--keystorePassphrase <keystore passphrase>]
[-kp|--keyPassphrase <key passphrase>]
[--keygenPassphrase <keygen passphrase>]
[-sp|--securityProvider <Java Security Provider>]
[-kst|--keystoreType <keystore type>]
[-cp|--cipherProvider <cipher provider>
[-ksl|--keystoreLocation <keystore location>]
[-kl|--keyLength <key length>]
```

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

To run the command use the following syntax:

On a Windows operating system:

```
dx-keytool.bat -c <command> -u <user> -p <password> <additional options as applicable>
```

On a UNIX operating system:

```
dx-keytool.sh -c <command> -u <user> -p <password> <additional options as applicable>
```

The following table describes the Data Integration Hub Security Keytool command line API options and arguments:

Option	Argument	Description
-c --command	command	Required. Command to run. Enter one of following commands: - rollKey. Rolls the master key. - rollPassphrases. Rolls the keystore passphrase and key passphrase. - rollConfig. Rolls the configuration. - generateKey. Generates a master key. - testConfig. Tests the current security configuration.
-u --user	user ID	Optional. User ID of User ID of a Data Integration Hub Administrator user account. If you use Informatica domain authentication or Informatica domain with Kerberos authentication, the user ID must specify the Informatica security domain, separated by the @ symbol. For example: `Administrator@SecurityDomain`
-U	environment variable	Optional. Environment variable that contains the value of userID. User ID of a Data Integration Hub Administrator user account. 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 ID options, -u or -U.
-p --password	password	Optional. Password of the Data Integration Hub Administrator user account. Enter a clear text password.

Option	Argument	Description
-P	environment variable	Optional. Environment variable that contains the value of password. Password of the Data Integration Hub Administrator user account. The password that you specify as a value of this environment variable must be encrypted. **Note:** You must specify at least one of the password options, -p or -P.
--server	hostname:port	Optional. Host name and port number of the Data Integration Hub server. If you do not enter a value, the API connects to the localhost server with the default port 18095. You must enclose the value in quotation marks. For example: `dx-keytool --server "localhost:18095"`
-old_ksp --oldKeystorePassphrase	current keystore passphrase in commands where `-ksp` is used for the new keystore password	Required for the rollPassphrases command. Optional for all other commands. Current passphrase to access the keystore. By default: `default`.
-old_kp -- oldKeyPassphrase	current key passphrase in commands where `-kp` is used for the new key password	Required for the rollPassphrases command. Optional for all other commands. Current passphrase to access the encryption key. By default: `default`
-ksp --keystorePassphrase	keystore passphrase	Required for the rollPassphrases and rollKey commands. Optional for all other commands. Passphrase to access the keystore. By default: `default`
-kp --keyPassphrase	key passphrase	Required for the rollPassphrases and rollKey commands. Optional for all other commands. Passphrase to use to access the encryption key. By default: `default`
--keygenPassphrase	key generator passphrase	Required for the rollKey command. Optional for all other commands. New passphrase to use to generate the master key.
-sp --securityProvider	security provider	Required for the rollKey command when you change the security provider. Optional for all other commands. Java security provider. The following conditions must exist before you run the command: - The provider must exist in the Java security configuration. For details, see https://docs.oracle.com/cd/E19830-01/819-4712/ablsc/index.html. - A custom provider JAR file must be part of the `DX server` classpath.
-kst --keystoreType	keystore type	Required for the rollConfig command when you change the keystore type. Optional for all other commands. Keystore type. The name of the keystore type must be compatible with the security provider that you want to use.

Option	Argument	Description
-cp --cipherProvider	cipher provider	Required for the rollConfig command when you change the cipher provider. Optional for all other commands. Cipher provider. The provider must exist in the Java security configuration.
-ksl --keystoreLocation	keystore location	Optional. Keystore location.
-kl --keyLength	key length	Optional. Can be used with the rollKey command. Key length in bits. Supported values: 128, 192, 256 Default value: 128

Individual Command Syntaxes

The following list describes the syntax that you use to run each of the Data Integration Hub Security Keytool Command Line API commands.

Roll the master key

```
dx-keytool.bat -c rollKey -u <user> -p <password> -ksp <current keystore passphrase>
-kp <current key passphrase> --keygenPassphrase <new key generator passphrase>
```

Roll the keystore passphrase and key passphrase

```
dx-keytool.bat -c rollPassphrases -u <user> -p <password> -ksp <new keystore
passphrase> -kp <new key passphrase> -old_ksp <cutrrent keystore passphrase> -old_kp
<current key passphrase>
```

Roll the configuration

```
dx-keytool.bat -c rollConfig -u <user> -p <password> sp <security provider> -kst
<keystore type> -cp <cipher provider> -ksp <current keystore passphrase> -kp
<current key passphrase>
```

Generate a master key

```
dx-keytool.bat -c generateKey -u <user> -p <password> -ksp <current keystore
passphrase> -kp <current key passphrase> --keygenPassphrase <new key generator
passphrase>
```

Test the current security configuration

```
dx-keytool.bat -c testConfig -u <user> -p <password>
```

Mask Sensitive Data

This section describes how to mask sensitive data in Data Integration Hub.

The Data Integration Hub administrator can integrate Informatica Dynamic Data Masking with Data Integration Hub to apply a security policies on specific topic table fields. Enabling dynamic data masking ensures that users cannot retrieve sensitive data. For example, you can mask social security number or credit card number in data that Data Integration Hub processes. For more information about using Dynamic Data Masking, see the *Informatica Dynamic Data Masking User Guide*.

Note: Ensure that you have installed Dynamic Data Masking. For more information about suported platforms, see *Informatica Dynamic Data Masking Installation Guide*

Perform the following tasks to mask sensitive data in Data Integration Hub.

1. Configure Dynamic Data Masking as follows:
   - Create a Dynamic Data Masking Service in the Dynamic Data Masking Management Console.
   - Define database connection properties for the database that requires data masking.
   - If the publishing database is Oracle, add a Transparent Network Substrate (TNS) entry in the `tnsnames.ora` file.
   - Create a connection rule to switch all incoming connections to the database.

   For more information about configuring Dynamic Data Masking, see "Configuring Dynamic Data Masking" on page 26.
2. Configure the Data Integration Hub publication repository to connect to the Dynamic Data Masking proxy server. For more information about configuring Data Integration Hub see, "Configuring Data Integration Hub to Activate Data Masking Rules " on page 28.
3. Create a subscription within an application and reference the name of the application in the data masking security ruleset. For more information about creating a subscription, see "Masking Data in Data Integration Hub" on page 28.
4. Define data masking rules in Dynamic Data Masking.
   - Create a security ruleset. Define rules to mask data.
   - Append another rule to the connection rule and assign the security ruleset to the connection rule.

   For more information about creating connection rules and security ruleset, see "Configuring Data Masking Rules in Dynamic Data Masking" on page 28.

Integrating Data Integration Hub with Dynamic Data Masking

The Data Integration Hub administrator integrates Data Integration Hub with Informatica Dynamic Data Masking, to mask sensitive data selectively. Dynamic Data Masking acts as a proxy server for the Data Integration Hub publication repository server. Subscription data access requests to Data Integration Hub go through the Dynamic Data Masking proxy server. The server masks data according to rules defined in Dynamic Data Masking and returns results to the subscription target.

Perform the following steps to integrate Data Integration Hub with Dynamic Data Masking.

- "Configuring Dynamic Data Masking" on page 26
- "Connecting Data Integration Hub to Dynamic Data Masking" on page 28

Configuring Dynamic Data Masking

This topic describes how to configure Dynamic Data Masking to integrate with Data Integration Hub.

To perform steps provided in this procedure, ensure that you have the required licenses and have installed Dynamic Data Masking.

For more information about adding services and configuring database connections, see the *Informatica Dynamic Data Masking Administrator Guide*.

Perform the following steps in the Dynamic Data Masking Management Console to connect Dynamic Data Masking with Data Integration Hub:

1. Add a Dynamic Data Masking service in Dynamic Data Masking.

2. Add a database and define database parameters that the Dynamic Data Masking service uses to connect to the target database. The target database configuration specifies the database, user account, and connection settings of the database on which the Data Integration Hub publication repository is installed.
   a. In the Management Console, click **Tree** > **Add Database**.

      The Add Database Window displays.

   b. Enter database details and save the information.
3. If Data Integration Hub uses Oracle server as a publishing database, add a TNS entry in `tnsnames.ora` file with details that refer to the Dynamic Data Masking proxy service that you have created in the previous step.
   An example of the TNS entry is as follows:

```
ORCL =

   (DESCRIPTION =

   (ADDRESS = (PROTOCOL = TCP)(HOST = <DDM_server_machinename>)(PORT = 1525))

   (CONNECT_DATA =

   (SERVER = DEDICATED)

   (SERVICE_NAME = ORCL.informatica.com)

  )

)
```

4. Create connection rules and add connection rules to a ruleset as follows:
   a. In the Management Console, select the Dynamic Data Masking service that you want to add the connection rule to, and click **Tree** > **Connection Rules**.

      The **Rule Editor** window is displayed.

   b. Select **Action** > **Apend Rule**.

      The **Apend Rule** window is displayed.

   c. Enter the following details to create the first connection rule that directs the incoming connection to the desired database:
      - **Rule Name**. Enter the name of the rule. For example, Rule 1.
      - **Identify incoming connections using**. Select **Current Target Database**.
      - **Database**. Enter the name of the Data Integration Hub database.
      - **Apply Action on Incoming Connection**. Select **Switch to Database**.
      - **Database**. Enter the name of the database that you configured in the "Connecting Data Integration Hub to Dynamic Data Masking" on page 28.
      - **Processing Action: When Rule is matched**. Select **Continue**.

   d. Click **OK**.

      Connection Rules are saved.
5. Click **Test Connection** to validate the connection to the database.
6. Click **OK** to save the configuration.

Connecting Data Integration Hub to Dynamic Data Masking

This topic describes how to connect Data Integration Hub with the Dynamic Data Masking tool.

1. In the Navigator, click **Hub Management** > **Connections** > **DIH_STAGING**.

   The **Edit Connection** window is displayed.
2. Update the following information depending on the database server that you have configured with Dynamic Data Masking and click **Save**.
   - If you configured an SQL server in the Dynamic Data Masking console, enter the **Connection String** in the following format: `hostname, port_number`.
   - If you configured an Oracle server in the Dynamic Data Masking console, enter **Server Name** in the following format: `ConnectionString_SID`. SID is the server identification name that you have configured in the Dynamic Data Masking service in the `tnsnames.ora` file.
3. To test a connection, click the **Test Connection** icon that is next to the connection that you want to test. Data Integration Hub tests the connection and shows a message with test results.

Masking Data in Data Integration Hub

Perform the following steps to mask data selectively in Data Integration Hub:

- "Configuring Data Integration Hub to Activate Data Masking Rules " on page 28
- "Configuring Data Masking Rules in Dynamic Data Masking" on page 28

Configuring Data Integration Hub to Activate Data Masking Rules

This section describes how to configure Data Integration Hub to mask data selectively.

This section provides an overview of tasks that you must perform to mask data in Data Integration Hub. For more information about each step, refer to the *Data Integration Hub Operator Guide*.

1. Create an application and add a subscription. For example, enter the name of the application as `DataMaskingApp`.
2. Use the name of the application in the Dynamic Data Masking security ruleset. To configure a dynamic data masking rule, refer to "Configuring Data Masking Rules in Dynamic Data Masking" on page 28.

   The subscription that you have created within the application is masked according to Dynamic Data masking rules.

Configuring Data Masking Rules in Dynamic Data Masking

This section describes how to configure security rules in Dynamic Data Masking. Data Integration Hub uses these rules to mask data that are requested by subscription. Security rules specify the technique that the Dynamic Data Masking rule engine uses to mask data. Security rules consist of a matcher, a rule action, and a processing action. Use security rules to mask data in a specific row or to mask an entire column. For example, you can create a security rule that rewrites SQL requests that reference the Social Security column from the Employee table.

For more information about creating security rules, see the *Informatica Dynamic Data Masking User Guide*.

Perform the following steps to define a security rule in Dynamic Data Masking:

1. Append a connection rule to the connection rules you created in the "Configuring Dynamic Data Masking" on page 26 procedure as follows:
   a. In the Management Console, select the Dynamic Data Masking service that you want to add the connection rule and click **Tree** > **Connection Rules**.

      The **Rule Editor** window is displayed.
   b. Select **Action** > **Apend Rule**.

      The **Apend Rule** window is displayed.
   c. Enter the following details to create the second connection rule that redirects to the data masking Security RuleSet that the connection rule needs to execute:

      Note: The first connection rule is created in "Configuring Dynamic Data Masking" on page 26.

      - **Rule Name**. Enter the name of the rule.
      - **Identify incoming connections using**. Select **All Incoming Connections**.
      - **Apply Action on Incoming Connection**. Select **Use Rule Set**.
      - **Rule Set Name**. Enter a name for the ruleset. You will further use the same name while you create the security ruleset in the next step. For example, `MaskEmpResultSet`.
      - **Processing Action: When Rule is matched**. Select **Continue**.
   d. Click **OK**.

      Connection Rules are saved.
2. Perform the following steps to create a ruleset.
   a. In the Management Console, select the Dynamic Data Masking service that you want to add the connection rule to, and click **Tree** > **Add RuleSet**.

      The **Add Ruleset** window is displayed.
   b. Enter the name of the security rule set that you gave as the Ruleset Name in the previous step and click **OK**. For example, `MaskEmpResultSet`.

      Within this rule set, create a rule or rules to match the column name in the result set that you want to mask, and specify the masking action.
   c. In the Management Console, click the security rule set that you created in the previous step.
   d. Select **Tree** > **Security Rule Set**.

      The Rule Editor window is displayed.
   e. Click **Action** > **Apend Rule** and update the following information:
      - **Rule Name**. Enter a name for the rule.
      - **Matching Method.** Select **Text**.
      - **Text Description**. Provide the name of the application that is defined in Data Integration Hub enclosed by the `%` symbol. For example, enter `%DataMaskingApp%`.
      - In the **Action** section, define masking rules as required.
      - **Action Type**. Select Mask
      - **Table Name**. Enter the name of the table that you have defined in the topic that maps to the application.
      - **Column Name**. Enter the name of the column that you want to mask. For example, `Credit Card Number`.

- **Masking Function**. Enter `****`

f. Click **OK** to save the ruleset.

The server masks data according to rules defined in Dynamic Data Masking and returns results to the subscription target. Considering the example values used in this procedure, credit card number of employees is replaced with **** when an application subscribes to the `DataMaskingApp` application.

CHAPTER 3

Events

This chapter includes the following topics:

Events Overview

An event is a representation of a publication or a subscription instance, at a particular stage of the publication or subscription process. The Data Integration Hub server generates events as it runs and processes publications and subscriptions, and it changes the status of the events as the publication or subscription process progresses. When an application triggers a publication, if the triggered publication has a pre-process, the publication event also tracks the pre-process. When a subscription triggers a post-process, the subscription event also tracks the post-process.

When an application that runs a publication pre-process publishes data or files to the publication repository, the Data Integration Hub server assigns an event to the publication as follows:

- If the pre-process passes the Publication event ID to the publication process, the publication uses the same event, and the Data Integration Hub server does not generate an additional Publication event for the publication process.
- If the pre-process does not pass the event ID to the publication process, the Data Integration Hub server generates another Publication event for the publication process.

If a file publication publishes more than one file, Data Integration Hub creates a File event for each file that it picks up. Data Integration Hub creates a Publication event after all the files are picked up.

The Publication event is the root event and the parent event for all of the subscription events that the Data Integration Hub server generates during the publication process. After the published data is ready for subscribers, the Data Integration Hub server generates a Subscription child event for each subscriber that needs to consume the published data. The Publication event contains aggregated status information for all Subscription child events.

An event changes status as it is processed. The Data Integration Hub server changes the status of an event based on a set of event statuses that you define in the Operation Console. As each subscriber consumes the

published data, the child event finishes processing. After all subscribers consume the published data, the Data Integration Hub server updates the consumption status of the parent event for the publication.

You can manage storage space in the Data Integration Hub run-time repository by purging events that Data Integration Hub generates when they are no longer needed.

Event Types

When the Data Integration Hub server runs and processes publications and subscriptions, it creates events according to the event types that are defined in the Operation Console.

In the Operation Console, you can create custom event types based on your processing requirements. You can then use the custom type in a custom PowerCenter workflow. You can edit or delete an event type that you create.

The event type indicates the type of action that took place in Data Integration Hub: publication, subscription, compound subscription, aggregated subscription, or file pick up. File pick up events and custom events do not appear in the Dashboard charts and reports.

The following table describes the default event types:

Event Type	Description
Aggregated Subscription	Event generated when multiple published data sets are ready to consume and you select to deliver the data sets with a single subscription mapping. The Aggregated Subscription event contains references to all Subscription events that were generated for each published data set. The Subscription events inherit their status from the Aggregated Subscription event.
Compound Subscription	Event generated when a subscription consumes published data from multiple publications. The event contains references to all Subscription events that were generated from each publication.
Custom Event	Predefined custom event. You can configure a workflow to generate events of this type. Used by the Data Integration Hub Server for archiving and monitor events.
Data-driven Publication	Event associated with a data-driven publication process. Each time an application publishes data with a data-driven publication, Data Integration Hub creates a Data-driven Publication event. The event acts as a parent event for all Subscription child events.
File Event	Event associated with a pickup of a file that needs to be published. If a file publication publishes more than one file, Data Integration Hub creates a File event for each file that it picks up. Data Integration Hub creates a Publication event after all the files are picked up. If not all files are found, the status of the publication event is set to Error.
Publication	Event associated with a publication process, with the exception of data-driven publications. Each time an application publishes data, Data Integration Hub creates a Publication event. The event acts as a parent event for all Subscription child events.
Subscription	Event associated with a subscription process. Each time an application finishes publishing data, Data Integration Hub creates a Subscription event for each subscribing application. The event acts as a child event for the Publication or Data-driven Publication event. After all subscribers consume the published data, Data Integration Hub updates the consumption status of the parent publication event.

Event Type	Description
System Event	Event generated by the Data Integration Hub server for system notifications. For example, the Data Integration Hub server generates a system event when a compound subscription cannot consume published data from all required publications.
Transaction Level Event	Event associated with transactions included in a document. You can configure a PowerCenter workflow to generate events of this type.

Managing Event Types

Use the Navigator to create, edit, and delete event types. If you define custom event types, you can build PowerCenter workflows that assigns the new event types to events that Data Integration Hub processes.

1. In the Navigator, click **Monitoring and Tracking** > **Event Types**.

   The **Event Types** page appears.
2. Choose to create, edit, or delete an event type.
   - To create an event type, click **New Event Type** and enter the name of the event type.
   - To edit an event type, click the **Edit** icon next to the event type that you want to edit and change the event type name. You cannot edit or delete the System Event type.
   - To delete an event type, click the **Delete** icon next to the event type that you want to delete and confirm the deletion.

Event Statuses

Event State and Event Statuses

Select an **Event State** and the corresponding event statuses are displayed. The available event states are **Final** and **Non-final**. The event status indicates the progress of the event while Data Integration Hub processes it.

The following table describes the event state and their corresponding event statuses:

Event State	Event Status Type	Description
Final	Completed	Indicates that the subscription instance finished running and that the subscribing application consumed all published data.
Non-final	Delayed	- For publications: Relevant for File events, for publications that publish multiple files. Indicates that the file that is related to the event is ready but that not all the files that the publication publishes are ready. When all the files that the publication publishes are ready, Data Integration Hub creates a Publication event, and the related File events inherit the status of the Publication event. You cannot run delayed File events. - For subscriptions: Indicates that the published data is ready but that the subscribing application did not start consuming the data. Relevant for subscriptions with a defined schedule and for subscriptions that are run manually or by an external trigger. You can run delayed subscription events from the **Subscriptions** tab of the **Application Details** page.
Final	Discarded	Assigned when the event is delayed and then discarded due to a processing rule, or discarded manually. The event can be discarded from any state. This is applicable for both publication and subscription.
Final	Error	Indicates that the subscription instance encountered errors and did not finish running.
Non-final	Pre-processing	Indicates that the publication's pre-process workflows is running.
Non-final	Processing	Indicates that the publication or subscription instance is running.
Non-final	Post-processing	Indicates that the publication's post-process workflows is running.
Final	Reprocessed	Assigned when the event processes again.

In the Operation Console, you can create custom event statuses and types based on your processing requirements. You can then use the custom status or type in a PowerCenter workflow that you use for a custom mapping in a publication or a subscription.

For example, you can create a status that reflects a unique process in your organization, such as Sent for Approval. You can then configure a PoweCenter workflow to set the event to that status until the approval is processed.

You can edit or delete any event status that you create. You can also manually edit an event and set the event status to any status in the Operation Console.

Each event status includes a state property. The state property represents the processing stage for the event. The state property value determines whether an event appears in the Dashboard charts and reports. For example, only events with a state value of `Error` appear in the Errors by Application chart and report.

When you create a user-defined event status, assign a value to the state property. Failure to assign state the correct value might result in incorrect Dashboard reports.

Managing Event Statuses

Use the Navigator to create, edit, or delete event statuses.

1. In the Navigator, click **Monitoring and Tracking** > **Event Status**.

   The **Event Statuses** page appears.
2. Choose to create, edit, or delete an event status.
   - To create an event status, click **New Event Status** and define the event status properties on the **Create Event Status** page.
   - To edit an event status, click the **Edit** icon next to the event status that you want to edit and change the event status properties on the **Edit Event Status** page.
   - To delete an event status, click the **Delete** icon next to the event status that you want to delete and confirm the deletion. You cannot delete a system-defined event status.

Event Status Properties

The event status indicates the processing stage for the event and whether the event encountered errors during processing.

The following table describes the event status properties:

Property	Description
Event Status Name	Name of the event status.
Icon	Optional. Image to display in the **Status** column when you view the event on the **Event List** page.
State	Optional. State of the event during processing. You can choose one or more of the following options: - Final. The event finished processing successfully. - Error. The event encountered an error. If you select Final and Error, the event status indicated that the event finished processing with errors. If you do not select a state property value, the event status represents an intermediate stage in event processing. You cannot edit publications or subscriptions until the event reaches a Final state.

Event Attributes

An event attribute is a parameter that you can associate with a custom workflow to collect business-related information when the associated workflow processes documents. The Data Integration Hub server saves the information from the workflow to the repository.

You create event attributes in the Operation Console based on the information that you want to collect about the event or the document. When you create a workflow for a custom mapping in Data Integration Hub, you select the event attribute to use in the workflow from the list of available event attributes.

You view the event attributes and the values in the **Event Details** section of the **Events** page in the Operation Console. When you perform an advanced search for events, you select from the list of available event attributes to search for an attribute value.

Event Attribute Properties

Event attributes are parameters that store additional information about the processed events. Use the **Event Attributes** page to view and manage event attributes.

The following table describes the event attribute properties:

Property	Description
Attribute Name	Name of the event attribute.
Description	Optional. Description of the event attribute.

Managing Event Attributes

Use the Navigator to create, edit, or delete event attributes.

1. In the Navigator, click **Monitoring and Tracking** > **Event Attributes**.

   The **Event Attributes** page appears.
2. Choose to create, edit, or delete an event attribute.
   - To create an event attribute, click **New Attribute** and define the event attribute properties on the **Create New Attribute** page.
   - To edit an event attribute, click the **Edit** icon next to the event attribute that you want to edit and change the event attribute properties on the **Edit Attribute** page.
   - To delete an event attribute, click the **Delete** icon next to the event attribute that you want to delete and confirm the deletion.

Publication and Subscription Event Types and Statuses

Data Integration Hub assigns a default set of event types and event statuses to publication and subscription events when the following conditions exist:

- A publication with a publication pre-process runs.
- A publication or a subscription with an automatic mapping runs.
- A publication or subscription with a custom mapping that is associated with a Data Integration task runs.
- A subscription post-process runs.
- A data-driven publication runs.

Note: It is recommended that publications and subscriptions with a custom mapping that is associated with a PowerCenter workflow use the same event statuses and types as those that Data Integration Hub assigns to the automatic mappings. The developer assigns event statuses and types for custom mapping in the PowerCenter workflow.

Default Event Types

Data Integration Hub assigns the following event types:

- Publication. Assigned to a publication process that is not initiated by a data-driven publication. Acts as the parent event for all Subscription events and for File events of publications that publish multiple files. For File events of publications that publish a single file, the event log includes a link to the file on the Data Integration Hub document store.
- Data-driven Publication. Assigned to a publication process of a data-driven publication.
- File Event. Assigned to the publication of a file in publications that publish multiple files. The event log includes a link to the file on the Data Integration Hub document store.
- Subscription. Assigned to a subscription process. Acts as a child event for a publication event. For events of subscriptions that consume pass-through files and do not use file transfer, the event log includes a link to the file on the Data Integration Hub document store. For events of subscriptions that use file transfer to consume files, the event log includes a link to the file transfer list.
- Aggregated Subscription. Assigned to a subscription process that consumes multiple data sets from the same topic with a single subscription mapping. The event contains references to all Subscription events that were created when the associated topic finished publishing each data set. The Subscription events inherit their status from the Aggregated Subscription event.
- Compound Subscription. Assigned to a subscription process that consumes data sets from multiple topics with a single subscription mapping. The event contains references to all Subscription events that Data Integration Hub creates when each topic publication finished publishing the data set.

Default Event Statuses

For publications, Data Integration Hub assigns the following event statuses:

- Pre-processing. Indicates that the publication pre-processing instance is running.
- Processing. Indicates that the publication instance is running.
- Delayed. Relevant for File events, for publications that publish multiple files. Indicates that the file that is related to the event is ready but that not all the files that the publication publishes are ready. When all the files that the publication publishes are ready, Data Integration Hub creates a Publication event, and the related File events inherit the status of the Publication event. You cannot run delayed File events.
- Completed. Indicates that the publication instance finished running and that the data is ready for subscribers.
- Error. Indicates that the publication instance encountered errors and did not finish running.

Each Publication event also shows the consumption status of the child Subscription events. The status reflects the overall consumption and changes after all Subscription events changed status. For example, the consumption status changes to complete after all subscribers finished consuming the published data.

For subscriptions, Data Integration Hub assigns the following event statuses:

- Delayed. Indicates that the published data is ready but that the subscribing application did not start consuming the data. Relevant for subscriptions with a defined schedule and for subscriptions that are run manually or by an external trigger. You can run delayed subscription events from the **Subscriptions** tab of the **Application Details** page.
- Processing. Indicates that the subscription instance is running.
- Completed. Indicates that the subscription instance finished running and that the subscribing application consumed all published data.
- Post-processing. Indicates that the subscription post-processing instance is running.
- Error. Indicates that the subscription instance encountered errors and did not finish running.

Event Purging

Manage storage space in the Data Integration Hub document store by purging events that Data Integration Hub generates when they are no longer needed.

Data Integration Hub purges files that are saved to the document store with File events that it generates for flat file publications. Data Integration Hub does not purge files that are saved with File events that it generates for pass-through file publications.

You purge events with Informatica Lifecycle Management Data Archive, by using short term archive projects.

Event Archiving Process with Data Archive

Use Data Archive to define archive projects and schedule standalone and recurring archive jobs in Data Integration Hub or Data Archive.

The archive process with Data Archive typically includes the following stages:

1. Install and configure the Data Integration Hub accelerator in Data Archive. The accelerator accesses the Data Integration Hub repository and selects events for purge based on the archive project settings that you define in Data Archive. For installation instructions, see the *Data Integration Hub Installation and Configuration Guide*.
2. Configure the source connection properties to the Data Integration Hub repository.
3. Set up user roles and configure secured access. Create access roles and assign security groups to the source archive locations to determine which users can run archive jobs in Data Archive. You must create an access role even for a single user.
4. Create and publish the archive project. In the archive project, you define the parameters according to which you want the archive job to purge the events from the source purge location. For example, you can define to purge events of a specific state or application. Although you can only purge whole event hierarchies, some archive parameters only apply to the root event in the hierarchy.
5. Schedule and run the archive job. The archive job uses the parameters from the archive job that you create and purges the events and documents from the source location. You can schedule a single immediate job or a recurring job.
6. Periodically, rebuild the indexes of the Data Integration Hub repository schema. You can use the `ALTER INDEX <index name> REBUILD ONLINE` syntax.

Event Archiving with Data Archive Rules and Guidelines

When you use Data Archive to purge events from Data Integration Hub, consider the following rules and guidelines:

- You must enable the Data Archive process to access the locations that you define in the source connections.
- If you move the document store, you must update the location in the **Source / Staging Attachment Location** property for the source connection.
- When you create the archive project, you must select **Purge Only**.
- The archive jobs do not delete documents that Data Integration Hub stores in temporary folders of the Data Integration Hub document store. If you no longer require access to the files in the temporary folder, manually delete these files. Do not delete documents that Data Integration Hub creates on the day that you delete the documents.

Event Purge Script

Use the purge batch script to delete the events in the Data Integration Hub repository. You must have the Archiving privilege to purge events.

For more information on privileges, see the Administrator Role Privileges section.

You can find the purge script (purge.sh or purge.bat) in the following directory:

<DIH_HOME>\dx-tools\

In the Data Integration Hub repository utility, run the purge.bat or purge.sh command with the following syntax in a single line:

```
purge.bat/sh -d <days> [-excludesystemevents] [-p <password>] [-P <password>] [--server
<hostname:port>] [-u <loginName>] [-U <loginName>]
```

The following list describes parameters and arguments for the purge scripts.

-u or --user

Argument: userID.

Optional. Identifier of the Operation Console user account to archive the logs stored in the Data Integration Hub repository.

The user account must have Archiving privilege to all the data in the repository.

If you use Informatica platform authentication, the user ID must specify the Informatica security domain, separated by the @ symbol. Informatica security domain, separated by the @ symbol.

```
Administrator@SecurityDomain
```

Note: You must specify at least one of the user name options, -u or -U.

-U

Argument: Environment variable.

Optional. Environment variable that contains the value of UserID.

Identifier of the Operation Console user account must have the Archiving privilege.

If you use Informatica platform authentication, the user ID 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 or --password

Argument: password.

Optional. Password of the Operation Console user that executes the archive command. This option contains the clear text password.

You must specify at least one of the password options, -p or -P to determine the user's password required to execute this command.

-P

Argument: Environment variable.

Optional. Environment variable that contains the value of the password.

Password for the Operation Console user that executes the archive command. The password must be encrypted.

Note: You must specify at least one of the password options, -p or -P.

--server

Argument: “<hostname:port>”

Optional. Host name and port number of the Data Integration Hub server. If you do not enter a value, the archive utility connects to the localhost server with the default port 18095. You must enclose the argument in quotation marks. For example:

```
purge --server “localhost:18095”...
```

--excludesystemevents

Argument: Exclude System Events

Optional. The system events are purged if the parameter is not specified. The system events are ignored if the parameter is specified.

-d

Argument: days.

The number of days before which the event purge occurs. The value can be greater than or equal to the value mentioned for dih.staging.max.lifetime.

Short-Term Event Archiving Connection Properties

Before you create and run archive jobs from the production database, you configure source connection properties for the production database.

The following table describes the production database source connection properties and values:

Property	Description	Value
Application Version	Version of the application.	Must match the installed Data Integration Hub version.
Source / Staging Attachment Location	Root location of the Data Integration Hub document store.	Must match the value of the dx.system.document.store.folder system property in Data Integration Hub.

Short-Term Event Archiving Access Roles Properties

Use the **Assign Role to Entity** section of the **Manage Access Roles** page in Data Archive to set up access roles to each archive location. After you create the access roles, you add the roles to security groups and assign the security groups to the source or target connections.

The following table describes the access role properties and the values to enter for short-term archiving:

Property	Description	Value
Application Version	The version of the application.	Data Integration Hub <version>
Application	The source archive location.	DIH_SCHEMA
Entity	The archive entity.	Processing Data

Archive Projects in Data Archive

You manage archive projects in Data Archive.

Before you create the archive project, you configure the source connections to the source location from which you want to purge events. You set up access roles and assign security groups to the source locations to determine which users can run archive jobs in Data Archive.

When you create the archive project, you define project parameters to control which events and documents to purge. The archive project contains required and optional parameters. Required parameters apply to the entire event hierarchy. Optional parameters apply to the root event in the event hierarchy.

For required parameters that apply to the entire event hierarchy, the archive job purges only event hierarchies in which all events match the parameter. For example, if you select to purge events older than 10 days, the archive job purges event hierarchies in which all events are older than 10 days.

For optional parameters that apply to the root event in the hierarchy, child events of the hierarchy do not need to match the parameter. For example, if you select to purge events with a related topic, the archive job purges event hierarchies in which only the root event has a related topic. The child events in the event hierarchy can have different related publications.

After you create the archive project, you schedule and run an archive job. The archive job contains the archive project that you created and any additional operations that Data Archive needs to perform to successfully complete the event purging.

Archive Project Parameters

When you create the archive project you define the parameters for the accelerator to use when the archive job runs.

The following table describes the archive project parameters:

Parameter	Description
Event Age (days)	Required. Minimum number of days from the event creation date for the entire event hierarchy. For example, if the event was created on March 1, 2014 and the date that the archive job runs in March 10, 2014, the age of the event at the time that the archive job runs is 10 days. Therefore, if the value of the parameter is higher than 10, the archive job does not archive the event. If you enter the value 0 in the parameter, the archive job purges all events that match the other parameter values regardless of the event age. **Note:** All events that Data Integration Hub generates from 00:00 until 23:59 on the same day have the same event age.
Application	Related application for the root event in the event hierarchy.
Topic	Related topic for the root event in the event hierarchy.
Publication	Related publication for the root event in the event hierarchy.
Event State	Event state for the root event in the event hierarchy. **Note:** If you do not specify a value, the archive job purges only event hierarchies in which all events reached a final state.

Creating an Archive Project

After you configure the source and set up user access to the archive locations, you create an archive project in Data Archive. In the archive project, you assign the source to the project, define user roles, and define the archive parameters that determine which events and documents to purge.

1. On the **Manage Archive Projects** page in Data Archive, create an archive project.
2. On the **General Information** page, enter a name for the project and select the **Purge** action.
3. Select the source.
4. Assign a user role to the archive project.
5. Define the archive parameters to determine which events and documents to purge.
6. On the **Manage Execution** page, choose whether to publish the project and run the archive job immediately or save the project and schedule the archive job separately.

Scheduling an Archive Job

After you create the archive project, you schedule an archive job that includes the archive project definitions. You can run the job immediately, at a later time, or as a recurring job.

1. On the **Schedule a Job** page, select the archive project that you want to run and add the program that you want to run.
2. Define the job schedule and click **Schedule**.

   If you scheduled the archive job to run at a later time, you can view the job status on the **Manage Jobs** page. After the archive job ends, you can view the job status and results on the **View Job History** page.

Event Monitors

The Data Integration Hub operator can create event monitors that track publications and subscriptions based on their event status, and instigate actions when an event is in a defined status.

The operator creates monitoring rules that define which entities to monitor, what are the event statuses for which to take action, and what actions Data Integration Hub takes when an event reaches the defined status.

For Data Integration Hub to send email notifications, you must edit some of the default monitor system properties. Optionally, you can customize the default email notification, for example, change the text in the body of the email.

Enabling and Customizing Email Notifications

Enable and customize email notifications that Data Integration Hub sends when the conditions of a monitoring rule which is configured to send email notifications are true.

1. In the Navigator, click **Administration** > **System Properties**.

   The **System Properties** page appears.
2. Enter values for the following properties:

dx.smtp.login

Login name of the Data Integration Hub SMTP server administrator account.

dx.smtp.password

Password of the Data Integration Hub SMTP server administrator account.

dx.smtp.port

Port number of the Data Integration Hub SMTP server.

dx.smtp.server

URL of the Data Integration Hub SMTP server.

For more information, see "Managing System Properties" on page 63.

3. If the Data Integration Hub SMTP server communicates through SSL, change the value of `dx.smtp.ssl` to `true`.
4. To customize the email notification, edit any of the following properties:

dx_email_body_file

Path to a template that contains the custom body of the email. The template must be a valid HTML file that can be compiled by an Apache Velocity engine. You can use the following placeholders in the template:

- $eventId
- $eventLink
- $publication
- $subscription
- $topic
- $application
- $eventCompleteDate
- $eventStatus
- $ruleName
- $description
- $ruleContent

dx_email_from_field

String that replaces the From field of the email.

dx_email_subject_field

Subject field of the email.

For more details, see "Event Monitor System Properties" on page 60.

CHAPTER 4

User Policies

This chapter includes the following topics:

User Policies Overview

User policies determine which users can log in to Data Integration Hub and access information that Data Integration Hub processes.

You manage user policies in the following areas of Data Integration Hub:

- User authentication. Credentials for Data Integration Hub user accounts. User authentication controls which users can log in to the Operation Console. You can use native authentication or Informatica domain authentication.
- User groups. Sets of users with permissions and privileges that determine the actions that users can perform in the Operation Console. You must include each user in one or more user groups.
- Categories. Additional permissions that you use to determine which user groups can access objects. You create categories and grant permissions for the categories to user groups. The operator assigns categories to applications and to topics. The administrator assigns categories to connections. The developer assigns categories to Data Integration Hub workflows.

User Authentication

User authentication determines the users that can log in to Data Integration Hub. The user authentication mode controls the location of user accounts and the tools for managing the accounts.

When you install Data Integration Hub you select a default administrator user name. The password is the same as the user name. Use the administrator account to manage user authentication in one of the following authentication modes:

- Data Integration Hub native authentication. Stores user accounts in the local Data Integration Hub repository. Use native authentication for a development or a test environment. When you install Data Integration Hub, the default password is the same as the user name. Use the Navigator of the Data Integration Hub Operation Console to manage users in the Data Integration Hub repository.
- Informatica domain authentication or Informatica domain with Kerberos authentication, depending on the authentication method that your Informatica domain uses. Synchronizes user accounts with the Informatica security domain. Use Informatica domain authentication or Informatica domain with Kerberos authentication for a production environment. Use the Administrator tool of the Data Integration Hub Operation Console to manage users. Use the Operation Console Navigator to synchronize users between the security domain and the Data Integration Hub repository and to assign user groups to users.

You can switch between Data Integration Hub native authentication and Informatica domain authentication or Informatica domain with Kerberos authentication, depending of the authentication method that your Informatica domain uses. If you switch from Data Integration Hub native authentication to Informatica domain authentication or to Informatica domain with Kerberos authentication, the user synchronization process overrides existing user accounts.

User Account Properties

You define user account details and user group assignments on the **Edit/Create User** page of the Operation Console. In Informatica domain authentication mode, user account details appear in read-only mode and you can define only user group assignments.

The following table describes the user account properties on the **Details** tab:

Property	Description
User ID	Unique identifier for the user account. In Native authentication mode, the user ID and password are identical. You cannot change this property after you create the user account. The maximum character length is 80.
Full Name	Name or description for the user account. The maximum character length is 255.
Email	Optional. Email address for the user account. The maximum character length is 255. **Note:** Data Integration Hub uses the email address you define here to send email notification from event monitors. If you do not define an email address for a user, and you define the user as a recipient of monitor notifications, Data Integration Hub does not send a notification when the conditions of the monitoring rule are true. For more information, see "Event Monitors" on page 42.

The following table describes the user account properties on the **User Groups** tab:

Property	Description
Available User Groups	List of user groups to which you can assign the user account.
Selected User Groups	List of user group assignments for the user account.

Managing Users in Native Authentication

Manage users on the **Users** page of the Operation Console. When you use native authentication, you manually define a user name and password for users in the Operation Console.

1. In the Navigator, click **Administration** > **Users**.

   The **Users** page appears.
2. Choose to create, edit, or delete a user.
   - To create a user, click **New User** and configure the user properties and user group assignments.
   - To edit an user, click the **Edit** icon next to the user that you want to edit and change the properties or user group assignments for the user. You cannot change the user ID property.
   - To delete an user, click the **Delete** next to the user that you want to delete and confirm the deletion.

You must assign the user account to one or more user groups to define permissions and privileges to the user. You cannot manage permissions for a single user account.

Switching to Native Authentication

Use the command line to switch from Informatica domain authentication or from Informatica domain with Kerberos authentication to Data Integration Hub native authentication.

1. In the command line, change to the following directory:

   ```
   <DIH_Install_Directory>/dx-tools
   ```
2. Run the following command:

   ```
   repoutil

   -c migrateToNative

   -l "<JDBC_URL_for_DIH_Repository>"

   -u <DIH_Repository_User_Name>

   -p <DIH_Repository_Password>

   --sysadmin <DIH_System_Administrator_User_Name>

   -t dih
   ```

   The authentication mode changes to native authentication and the system creates the administrative user that you specified in the --sysadmin command.
3. Use the Navigator to create additional users.

Managing Users in Informatica Domain Authentication

When you use Informatica domain authentication or Informatica domain with Kerberos authentication, you can synchronize users in the Informatica security domain with Data Integration Hub. Use the Operation Console Navigator to synchronize users between the security domain and Data Integration Hub. You synchronize users after you switch from Native authentication or if user information in the security domain changed.

To synchronize users in the Informatica security domain with Data Integration Hub, the following conditions must be true:

- The Informatica security domain is configured on the **Security** page of Informatica Administrator.
- At least one security group in the Informatica security domain contains the Data Integration Hub users to synchronize.
- The Data Integration Hub system property `dx.authentication.groups` contains the list of groups from the Informatica security domain to synchronize, in the following format:

  ```
  <group name>@<security domain> [;<groupname>@<security domain>]
  ```
- One of the groups that are defined in `dx.authentication.groups` contains the user that performs the synchronization.
- The user that is defined in the Data Integration Hub system property `pwc.repository.user.name` has privileges to manage users, groups, and roles.
- The Data Integration Hub user has privileges to synchronize users.

1. In the Navigator, click **Administration** > **Users**.

   The **Users** page appears.
2. Click **Synchronize Users**.

   The **Synchronize Users** page appears.
3. Click **OK** to synchronize users.
4. On the **Edit User** page, verify user group assignments for the users that you synchronize.

Switching to Informatica Domain Authentication

Use the command line to switch from Data Integration Hub native authentication to Informatica domain authentication.

1. In the command line, change to the following directory:

   ```
   <DIH_Install_Directory>/dx-tools
   ```
2. Run the following command:

   ```
   repoutil

   -c migrateToISP

   -l "<JDBC_URL_for_DIH_Repository>"

   -u <DIH_Repository_User_Name>

   -p <DIH_Repository_Password>

   -Ddx.pwc.domain.gateway=<PowerCenter_GatewayHost>:<PowerCenter_GatewayPort>

   -Ddx.pwc.user=<PowerCenter_User_Name>@<Security_Domain>

   -Ddx.pwc.password=<PowerCenter_Password> -t dih
   ```

The authentication mode changes to Informatica domain and the repoutil deletes all users from the Data Integration Hub repository.

3. Restart the Data Integration Hub Operation Console.
4. Synchronize the users from the Informatica platform security domain to the Data Integration Hub repository.

   For more information, see "Managing Users in Informatica Domain Authentication" on page 47.

Switching to Informatica Domain with Kerberos Authentication

Switch from Data Integration Hub native authentication or from Informatica domain authentication to Informatica domain with Kerberos authentication.

1. Create a `.properties` file with the following properties:
   - dx.kerberos.initial.administrator=<Kerberos_Domain_System_Administrator_Credentials>
   - dx.kerberos.krb5.file=<File_That_Stores_Keberos_Configuration_Information>
   - dx.kerberos.console.keytab.file=<Location_of_the_Keytab_File_for_the_Operation_Console>
   - dx.kerberos.console.service.principal.name=<SPN_for_the_Operation_Console>
   - dx.pwc.domain.gateway=<PowerCenter_Domain_Gateway_Host_and_Port_Number>
2. Optionally, add the following properties to the file:
   - dx_system_property.pwc.repository.user.name=<User_to_Access_the_PowerCenter_Repository_Service>
   - dx_system_property.pwc.repository.password=<Plain_Text_Password_to_Access_the_PowerCenter_Repository_Service>
   - dx_system_property.pwc.user.name.space=<PowerCenter_Security_Domain>

   For example, if the credentials changed during the upgrade. Data Integration Hub stores the password in an encrypted form.
3. In the command line, change to the following directory:

   ```
   <DIH_Install_Directory>/dx-tools
   ```
4. Run the following command:

   ```
   repoutil

   -c migrateToISPKerberos

   -l "<JDBC_URL_for_DIH_Repository>"

   -u <DIH_Repository_User_Name>

   -p <DIH_Repository_Password>

   -t dih
   --file <Properties_File_You_Created_in_Step_1>
   ```

   For example:

   ```
   repoutil.bat -c migrateToISPKerberos -l "jdbc:informatica:oracle://
   machine1:1521;SID=orcl" -u DIH_DB_USER -p DIH_DB_PASSWORD -t dih --file
   c:\migrateToISP.properties
   ```
5. If you performed step 2 and added the credential to the properties file, run the following command:

   ```
   repoutil

   -c loadProperties
   ```

```
--file <Properties_File_You_Created_in_Step_1>

-u <DIH_Repository_User_Name>

-p <DIH_Repository_Password>

-t dih
-l "<JDBC_URL_for_DIH_Repository>"
```

For example:

```
repoutil -c loadProperties --file c:\PowerCenter.properties -u DIH_DB_USER -p
DIH_DB_PASSWORD -t dih -l "jdbc:informatica:oracle://machine1:1521;SID=orcl"
```

The authentication mode changes to Informatica domain with Kerberos and the repoutil deletes all users from the Data Integration Hub repository.

6. Synchronize the users from the Informatica security domain to the Data Integration Hub repository.

User Groups

A user group defines permissions and privileges for Data Integration Hub user accounts. Permissions control the objects and data that users can access and monitor. Privileges control the actions that users can perform on objects.

Data Integration Hub contains the following default user groups:

- Administrator
- Developer
- Operator
- SysAdmin

You must assign each user to one or more user groups. You cannot edit or delete the default user groups. You can create, edit, and delete additional user groups.

When you create a user group, you assign one or more roles to the user group. Each role defines permissions to access and monitor defined types of data and privileges to use defined system functions. You can assign some or all of the permissions and privileges to the user group. The roles are predefined and you cannot create custom roles.

User Group Permissions

You define user group properties on the **Create/Edit User Group** page.

You define permissions to determine the object categories that the users in the group can view or change. When you assign categories to applications, to topics, and to connections, only user groups with permissions to the categories can view or change the objects.

The associated topic, application, or connection, whichever is the most restrictive, determines user permissions to publications and to subscriptions. For example:

- To create and to edit publications, users must have write permissions to both the associated application and the associated topic, and read permissions to the associated connection.
- To create and to edit subscriptions, users must have write permissions to the associated application and read permissions to all of the associated topics and to the associated connection.

The **Permissions** tab includes the following properties:

User Group Name

Textual name for the user group.

Category permissions

Determines whether to grant the user group read and write permissions to all categories or to specific categories. You can choose from the following options:

- Grant read and write permissions to all categories
- Select specific categories to grant read permissions or both read and write permissions

If you select specific categories, users in the group can access only objects with the selected categories and objects with no category assignments.

User Group Privileges

You define user group properties on the **Create/Edit User Group** page.

You define privileges to determine which actions users in the group can perform on different object types. To assign privileges, you select roles and add or remove privileges as needed.

Data Integration Hub contains the following roles:

- Administrator
- Developer
- Operator

You select roles and set privileges to the roles on the **Privileges** tab. Each role includes default privileges and some privileges appear in more than one role.

If you create an event type, Data Integration Hub assigns viewing privileges for that event type to the default user groups. To grant custom user groups access to the event type, you must manually assign viewing privileges to additional user groups that you create.

Administrator Role Privileges

The following table describes the actions that the Administrator role can perform on objects in Data Integration Hub:

Object	Action
Category	- View - Create - Edit - Delete
Connection	- View - Create - Edit - Delete
Events	- View
System Property	- View - Create - Edit - Delete

Object	Action
User	- View - Create - Edit - Delete
User Group	- View - Create - Edit - Delete
View Events	- System Event
Other	- Archiving - Export Data - Import Data - Synchronize Users - Use Key Aliases

Developer Role Privileges

The following table describes the actions that the Developer role can perform on objects in Data Integration Hub:

Object	Action
Connection	- View - Create - Edit - Delete
Event Attribute	- View - Create - Edit - Delete
Event Status	- View - Create - Edit - Delete
Event Type	- View - Create - Edit - Delete
System Property	- View - Create - Edit - Delete
Topic	- View

Object	Action
Workflow	- View - Edit
Other	- Use Key Aliases

Operator Role Privileges

The following table describes the actions that the Operator role can perform on objects in Data Integration Hub:

Object	Action
Application	- View
Category	- View - Create - Edit - Delete
Connection	- View - Create - Edit - Delete
Dashboard	- View
Event Attribute	- View
Event Status	- View
Event Type	- View
Event	- View
Monitor	- View - Create - Edit - Delete
Publication	- View - Create - Edit - Delete
Subscription	- View - Create - Edit - Delete
System Property	- View

Object	Action
Topic	- View - Create - Edit - Delete
User	- View
User Group	- View
View Events	- Custom Event - File Level Event - Group Level Event - Segment Level Event - System Event - Transaction Level Event - Publication - Subscription - Compound Subscription - Aggregated Subscription - Data-driven Publication - File Event
Workflow	- View
Other	- Catalog - Change Event Status - Reprocess Event - Run Publication/Subscription - Use Key Aliases

Managing User Groups

Manage custom user groups on the **User Groups** page of the Operation Console.

1. In the Navigator, click **Administration** > **User Groups**.

   The **User Groups** page appears.

2. Choose to create, edit, or delete a custom user group.
   - To create a custom user group, click **New User Group** and configure the permissions and privileges.
   - To edit a custom user group, click the **Edit** icon next to the user group that you want to edit and change the permissions and privileges. You cannot edit or delete default user groups.
   - To delete a custom user group, click the **Delete** icon next to the user group that you want to delete and confirm the deletion.

If you assign user group permissions to specific categories, you assign categories to applications to link them with the user group.

Categories

A category controls access to applications, topics, connections, and workflows. You assign categories to user groups to determine the users that can view or change the entities. Entities without categories are accessible by all users.

A category can represent different departments or areas of interest within your organization. For example, you might have several applications in the customer service departments and other applications in the finance department. You can create a customer service category and a user group with privileges for customer service personnel. Assign the customer service category to the customer service user group to enable access for those users to all applications that are relevant to their work.

Categories also control access to dependent entities, such as events and data in the Dashboard reports.

When the operator creates applications and topics, when the administrator creates connections, and when the developer creates workflows, they assign permissions to the entities they create by selecting a category from the available categories. The categories that the operator or the developer assign to applications, topics, and workflows must match the categories that you assign to the user groups.

For more information about assigning permissions to applications and topics, see the *Data Integration Hub Operator Guide*. For more information about assigning permissions to workflows, see the *Data Integration Hub Developer Guide*.

Managing Categories

Create, edit, or delete categories on the **Categories** page of the Operation Console.

1. In the Navigator, click **Administration** > **Categories**.

   The **Categories** page appears.
2. Choose to create, edit, or delete a category.
   - To create a category, click **New Category**, enter the name for the category, and click the green checkmark next to the category.
   - To edit a category, click the **Edit** icon next to the category that you want to edit, change the name of the category, and click the green checkmark next to the category.
   - To delete a category, click the **Delete** next to the category that you want to delete and confirm the deletion.

     Note: If you delete a category for an object with no other assigned categories, the object becomes accessible by all users.

CHAPTER 5

Operation Console Management

This chapter includes the following topics:

Operation Console Management Overview

View Operation Console log files to monitor user access to the Operation Console and user activity in the Operation Console.

Viewing Access Logs

You can examine the access logs to track and analyze actions that users perform in the Operation Console.

To track and analyze the user actions on a specific day, check the `localhost_access_log.<date>.txt` log files at the following directory: `<DIHInstallationDir>/DataIntegrationHub/tomcat/logs`. The date is in the format `yyyy-mm-dd`.

To view expanded details about user activity that include login activity, check the `dx-console.log` files at the same directory: `<DIHInstallationDir>/DataIntegrationHub/tomcat/logs`.

CHAPTER 6

System Properties

This chapter includes the following topics:

System Properties Overview

System properties affect the configuration of Data Integration Hub. You can use system properties to change how Data Integration Hub operates and how it processes data. The installer creates and initializes the system properties.

If you installed the Dashboard and Reports component, see "Dashboard and Reports System Properties" on page 117 for a description of the dashboard and reports system properties.

If you use Informatica Intelligent Cloud Service to publish and subscribe to cloud applications, see "Cloud Connectivity System Properties" on page 79 for a description of the cloud connectivity system properties.

To create topic tables from sources that exist in Enterprise Data Catalog, you must configure Enterprise Data Catalog system properties. Fo more information about Enterprise Data Catalog system properties, see "Enterprise Data Catalog System Properties" on page 59.

Note: If the system property name ends with `.password`, Data Integration Hub stores the system property value as an encrypted string.

General System Properties

General system properties determine Data Integration Hub behavior, such as authentication mode and document store location.

The following table describes general system properties:

System Property	Description
dih.automap.instances.parameter.precision	The precision of the publication instance parameters `DXPublicationInstanceId`, `DXPublicationInstanceDates`, and `DXPublicationInstanceDatesSQL` in subscription mapping. By default 90,000. Precision is calculated at 60 characters for each publication. For aggregated subscriptions that consume more than 1,500 publications increase the precision accordingly.
dih.events.api.max.results	The maximum number of events the `events API` returns. The number includes the parent and the child events. Default is 2000. For more information about Events API, see *Developer Guide*.
dih.event.details.input.file.hide	Hides the content of the file of any File Event in the Event Details page. You can set the following values: - True. The file content is not displayed in the Event Details page. - False. The file content is displayed in the Event Details page. Default is False.
dih.file.publication.aggregation.delay	For file publication, the wait time in minutes before the publication file events are aggregated. The events are aggregated and triggered after this delay period. The wait time is applicable to all the file events that exist when aggregation is evaluated. Not applicable for publications with the scheduling option **When the file is ready to be published**.
dih.filetransfer.allpatterns.wait.timeout	For file transfer publications, time in minutes to wait from the time when the oldest event was created until all the files in the publication arrive. If all files do not arrive within the defined time, Data Integration Hub discards the File events that are related to the Publication event and changes the status of the Publication event to Error. By default 10 minutes.
dih.filetransfer.publication.aggregation.delay	For file transfer publications, time in seconds to wait until releasing Delayed file events for processing. By default 30 seconds. **Note:** Do not set a value that is lower than 30 seconds. Not applicable for publications with the scheduling option **When the file is ready to be published**.
dih.filetransfer.publication.poll.interval	For file transfer publications, time interval in minutes for Data Integration Hub to scan remote sources. If this property does not exist or is empty, Data Integration Hub scans remote sources every five minutes. The value of this property must be from 3 to 30 minutes.
dih.flatfile.publication.aggregation.batch.delay	For single-pattern flat file publications, time in seconds to wait from the first event until publication file events are aggregated. The value of this property must be from 0 to 86400 seconds.

System Property	Description
dih.flatfile.publication.evaluation.interval	For single-pattern flat file publications, time interval in seconds for aggregation rule evaluation. The value of this property must be one of the following values: 10, 20, 30.
dih.help.hosted	Connects to hosted help. When set to False, Data Integration Hub connects to bundled help.
dih.hadoop.column.delimiter	Column delimiter in files that Data Integration Hub writes to and reads from a Hadoop publication repository. **Note:** If you change the value of this property you must open all publications that write to a topic with a Hadoop publication repository and all subscriptions that read from topic with a Hadoop publication repository in the publication or the subscription wizard, correspondingly, and run the wizard for each publication and subscription.
dih.hadoop.service.url	URL for the service that manages write and read activity for the Data Integration Hub Hadoop publication repository.
dih.home.hide.at.startup	Determines the display of the Operation Console Home page when users log in to Data Integration Hub. You can set the following values: - False. The Home page shows the Data Integration Hub Overview diagram. - True. The Home page shows a Welcome page. Default is False.
dih.realtime.time.window	For real-time publications, defines the buffer time that Data Integration Hub waits before the written publications are published. The default value is 1, which has no impact on the publication time. Change the buffer time when you have multiple tables in a topic and you want to ensure that the records for all tables are included in the same publication.
dih.splash.message	Define a splash message to display to all users. You can announce features, write a message on downtime or alerts, and so on. The maximum character length is 200.
dih.staging.lifetime.maximum	Maximum number of days to store published data in the publication repository. After the lifetime period for a publication instance expires, the publication repository cleanup process deletes the data from the publication repository, regardless of whether the data was consumed or not. Default is 90 days. **Note:** The value of dih.staging.lifetime.maximum must be larger than the value of dx.retention.period.default.days and larger than the values of the publication data retention period of any of the existing topics.

System Property	Description

dih.subscription.skip.staging.check	Based on the instance ID, a check is run on the publication repository for the number of rows published to the mapped tables of the topic.
	You can set the following values:
	- True. A check is not run on the publication repository for the number of rows published to the mapped tables of the topic.
	- False. A check is run on the publication repository for the number of rows published to the mapped tables of the topic.
	Default is False.
	Note:
	- The dih.subscription.skip.staging.check system property is applicable for the following subscription types:
	  - Automatic Relational Subscription
	  - Automatic Flat File Subscription
	  - Automatic Pass-Through Subscription
	- The dih.subscription.skip.staging.check system property is not applicable if the Join tables are mapped in Field Mapping.

dx.authentication.groups	Names of the Informatica domain security groups to synchronize. If you do not specify a value, the Informatica domain security domain synchronizes only the administrator user account that you define when you install Data Integration Hub. You can define multiple security groups separated by semicolons. You must enter the authentication group name and domain name in the following format: `<group>@<InformaticaSecurityDomain>[;<group>@<InformaticaSecurityDomain>]*` For example: B2B@Native
dx.authentication.mode	Active authentication mode.
dx.retention.period.default.days	Default value for the number of days to store published data in the publication repository after the published data is consumed. After the data retention period for a publication instance expires, the publication repository cleanup process deletes the published data set from the publication repository. The default value is seven days. The Data Integration Hub operator can change the data retention period in the **Data Retention** page of the **Topic** wizard. **Note:** The value of dx.retention.period.default.days must be smaller than the value of dih.staging.lifetime.maximum.
dx.retention.period.running.time	The time of day on which to run the publication repository cleanup. When the cleanup starts, Data Integration Hub checks the retention period for all published data sets in the publication repository and deletes published data sets for which the retention period expired before the cleanup started. Default is 23:00.

Enterprise Data Catalog System Properties

Enterprise Data Catalog system properties define the information to access Enterprise Data Catalog.

The following table describes Enterprise Data Catalog system properties:

System Properties	Description
edc.login.username	User name of the Enterprise Data Catalog resource.
edc.login.password	Password of the user account of the Enterprise Data Catalog resource.
edc.url	URL of the Enterprise Data Catalog resource, in the following format: `http://<hostname>:<port>`
edc.supported.column.types	Comma separated class types of Enterprise Data Catalog. For example, `com.infa.ldm.etl.pc.OutputTransformationPort, com.infa.ldm.relational.Column`
edc.supported.table.types	Comma separated table types of Enterprise Data Catalog. For example, `com.infa.ldm.relational.Table`

Event Monitor System Properties

Event monitor system properties determine the structure and the content of event monitor emails. Data Integration Hub sends the email when the conditions of a monitoring rule which is configured to send email notifications are true.

The following table describes event monitor system properties:

System Property	Description
dx_email_body_file	Path to a template file that contains a custom body of the email. For more information, see "Enabling and Customizing Email Notifications" on page 42. If this property does not exist or is empty, Data Integration Hub uses the default email template.
dx_email_from_field	String that replaces the From field of the email. By default: `donotreply@informatica.com`
dx_email_mimetype	MIME type of attachments to the email message. Default is "text/html"; charset=UTF-8.
dx_email_subject_field	Subject field of the email. If this property does not exist or is empty, Data Integration Hub uses the default email format: `<monitor_name> Occurred on <event_completion_date>`
dx.smtp.login	Login name of the Data Integration Hub SMTP server administrator account.
dx.smtp.password	Password of the Data Integration Hub SMTP server administrator account.

System Property	Description
dx.smtp.port	Port number of the Data Integration Hub SMTP server.
dx.smtp.server	URL of the Data Integration Hub SMTP server.
dx.smtp.ssl	Determines whether the Data Integration Hub SMTP server communicates through SSL or not. - False: The Data Integration Hub SMTP server does not communicate through SSL. - True: The Data Integration Hub SMTP server communicates through SSL. By default: False.

PowerCenter System Properties

PowerCenter system properties determine PowerCenter domain and connection management.

The following table describes PowerCenter system properties:

System Property	Description
pwc.domain.gateway	Name of the Informatica security domain gateway. The value of this property appears in the following format: `host_1:port_1(;host_2:port_2)`
pwc.domain.name	Name of the domain where PowerCenter is installed.
pwc.integration.service.name	Name of the PowerCenter Integration Service to run batch workflows.
pwc.repository.jdbc.url	JDBC connection URL for the PowerCenter repository database. Applicable for PowerCenter repositories that are hosted on Oracle or Microsoft SQL Server databases.
pwc.repository.jdbc.name	Database name of the PowerCenter repository database. Applicable for PowerCenter repositories that are hosted on Oracle or Microsoft SQL Server databases.
pwc.repository.jdbc.password	Password for the PowerCenter repository database. Applicable for PowerCenter repositories that are hosted on Oracle or Microsoft SQL Server databases.
pwc.repository.host	Host name of the node that runs the PowerCenter Repository Service.
pwc.repository.password	Password for the PowerCenter Repository Service. **Note:** You must restart the Data Integration Hub Operation Console after you change the password for the PowerCenter Repository Service.
pwc.repository.port	Port number for the node that runs the PowerCenter Repository Service.
pwc.repository.service.name	Name of the PowerCenter Repository Service.

System Property	Description
pwc.repository.user.name	Name of the PowerCenter Repository Service user.
pwc.user.name.space	Name of the security domain that stores the PowerCenter repository user. If you use Informatica domain authentication, change the default to grant access to the PowerCenter repository. Default value is Native.
pwc.webservices.url	URL or Web address of the PowerCenter Web Services Hub. For example: `http://localhost:7333/wsh/services/BatchServices/DataIntegration` **Note:** - PowerCenter Web Services doesn't support load-balancer. If you create multiple Web Services Hubs for fail over, you must configure a load-balancer in Active-Passive mode such that the Data Integration Hub always sends requests to a particular Web Services Hub, even if the load-balancer is in between Data Integration Hub and Web Services Hub. - If a node that make the batch service calls fails, update the property to the URL of another node manually. Batch service calls does not function if you link to the load-balancer URL.

Big Data System Properties

Big data system properties define Data Integration Service and Model Repository Service properties for the use of Data Engineering Integration and Data Quality mappings with Data Integration Hub custom publications and subscriptions.

The following table describes big data system properties:

System Property	Description
dis.domain.name	Name of the Informatica domain.
dis.integration.service.name	Name of the Data Integration Service.
dis.mrs.service.name	Name of the Model Repository Service.
dis.repository.host	Host name of the node on which the Data Integration Service runs.
dis.repository.password	Password for the node on which the Data Integration Service runs.
dis.repository.port	Port number for the node on which the Data Integration Service runs.
dis.repository.user.name	Name of the user for the node on which the Data Integration Service runs.
dis.security.domain.name	Name of the Informatica security domain in which the Model Repository Service user is stored.

Apache Kafka System Properties

Apache Kafka system properties determine the Apache Kafka domain and the connection URL.

The following table describes Apache Kafka system properties:

System Property	Description
kafka.staging.broker.url	URL of the Apache Kafka server. Appears in the following format: `host: port`

Managing System Properties

Use the Navigator to create, edit, and delete system properties. If the system property name ends with `.password`, Data Integration Hub stores the password as an encrypted string and displays asterisks in the **Value** column.

1. In the Navigator, click **Administration** > **System Properties**.

   The **System Properties** page appears.
2. Choose to create, edit, or delete a system property.
   - To create a system property, click **New Property** and enter the name and default value for the property.
   - To edit a system property, click the **Edit** icon next to the system property that you want to edit and change the property name or default value.
   - To delete a system property, click the **Delete** icon next to the system property that you want to delete and confirm the deletion.

CHAPTER 7

Connections

This chapter includes the following topics:

Connections Overview

Connections define the location and the credentials of the data source or the data target to which Data Integration Hub connects when a source application publishes data, or when a target application consumes published data.

You define connections for the following types of automatic mapping publications and subscriptions:

- Relational database publications and subscriptions.
- File publications and subscriptions that use a Hadoop Distributed File System (HDFS) source or target.
- File publications and subscriptions that use file transfer.

Data Integration Hub uses the connection details to create the workflow for the publication or for the subscription, and to create the data source or the data target in the PowerCenter repository.

You set up and manage connections to the data sources and to the data targets in your organization. You can create a connection to use for multiple publications and subscriptions.

Data Integration Hub creates the connection in the PowerCenter repository when you save the connection. The connection remains in the PowerCenter repository until you delete the connection in Data Integration Hub.

Connection Types

You can use the following connection types to publish and deliver data with an automatic mapping:

- Relational database. Use this type of connection to publish data from a relational database or to deliver data to a relational database. Relational database connection types use ODBC or native database drivers to access data at run time, and they use JDBC to access metadata at design time.

  When you create a relational database connection, you can choose from the following types:

  - Microsoft SQL Server. Connects to databases though ODBC or through the native database driver.
  - Oracle. Connects to databases through the native database driver.
  - IBM DB2. Connects to databases through the native database driver.
  - PostgreSQL. Connects to databases through ODBC.
  - ODBC. Connects to databases through ODBC. Use this connection type to connect to other relational databases. For example, create an ODBC connection to a MySQL database.

  Before you create an ODBC connection, you must configure the JDBC drivers for the relational database. For more information about configuring the JDBC drivers, see "Configuring the JDBC Drivers for an ODBC Connection" on page 68.
- HDFS. Use this type of connection to publish data from HDFS and to deliver data to HDFS. A HDFS connection is a file system type connection.
- File transfer. Use this type of connection to publish data from remote servers and to deliver data to remote servers with file transfer. Data Integration Hub supports FTP, SFTP, and FTPS as data source and data target connections.
- Teradata. Use this type of connection to publish data from Teradata and to deliver to Teradata. A Teradata connection is a type of relational database management connection.

Test Connections

After you create a connection to a source or to a target, you can test the connection to verify that the connection settings are correct and allow valid connectivity.

The operator can use a connection that is not valid to create publications and subscriptions.

Note: When you test an ODBC connection, Data Integration Hub validates the metadata access but does not validate the data access for the ODBC connection.

Connections to the Data Integration Hub Repositories

Data Integration Hub creates the following connections:

- DIH_REPO. Connection to the Data Integration Hub repository.
- DIH_STAGING. Connection to the Data Integration Hub publication repository.

- DIH_STAGING_HADOOP. Connection to the Data Integration Hub Hadoop publication repository. The connection shows on the Connections page of the Operation console if you select the Data Integration Hub Hadoop Service component during system installation.

Relational Database Connection Properties

A relational database connection can include the following connection details:

- General connection properties.
- Authentication details. Credentials of the database user account.
- Metadata access connection details. Data Integration Hub uses the metadata access connection details to access the database schema information when the operator creates or edits a topic and adds tables from a database.
- Data access connection details. Data Integration Hub uses the data access connection details at run time to connect to the source application or to the target application.
- Permissions. User categories that have permissions to the connection.

Relational Database Connection General Properties

A relational database connection includes the following general properties:

Connection Name

Name of the connection. The name can contain a maximum of 64 characters. The name cannot contain spaces or other special characters except for the underscore.

Description

Optional description of the connection.

Connection Type

The connection database type.

To connect to IBM DB2 databases, set the `showSelectableTables` parameter to false.

Relational Database Connection Authentication Properties

A relational database connection includes the following authentication properties:

Use Trusted Connection

Microsoft SQL Server database. Indicates whether to use Windows authentication to access the database. The user name that starts the connection must be a valid Windows user with access to the Microsoft SQL Server database.

User

Name of the database user account. The database user must have read and write permissions to the database.

If you define a connection to an Oracle database to process BLOB, CLOB, or NCLOB data, the user must have permissions to access and create temporary tablespaces.

Note: If you choose the option **Use Trusted Connection**, do not enter a user name here.

Password

Password for the database user account. The password must use 7-bit ASCII encoding.

Note: If you choose the option **Use Trusted Connection**, do not enter a password here.

Database Name

Microsoft SQL Server database connected with native driver. Name of the database.

Default Schema Name

The name of the schema to which Data Integration Hub connects. You enter the default schema name only for an ODBC connection.

Note: You cannot edit the **User** and **Password** properties for the connections to the Data Integration Hub repositories, DIH_REPO and DIH_STAGING, on the **Connections** page. For information about how you change users for those connections, see " Managing User Credentials for System Connections" on page 77.

Relational Database Connection Metadata Access Properties

A relational database connection includes the following metadata access properties:

Use This Connection for Metadata Access

Enable metadata access to the connection.

Connection String

Database connection string that you can use to preview data and create automatic mappings.

Connection String for Metadata Access Properties

A connection string defines the database connection.

Use the following examples to define connection strings for Oracle and Microsoft SQL Server databases:

Oracle

Example syntax for an Oracle database:

```
jdbc:informatica:oracle://myhost:1521;SID=mysid;CatalogOptions=4;loginTimeout=30;
```

Note: To use Oracle synonyms, set the value of `CatalogOptions` to `7` or remove the CatalogOptions key from the connection string. For example:

```
jdbc:informatica:oracle://myhost:1521;SID=mysid;CatalogOptions=7;loginTimeout=30;
```

If you do not want to use the default schema to which Data Integration Hub connects, append the following string to the connection string:

```
ALTER SESSION SET CURRENT_SCHEMA=<schema name>
```

where `<schema name>` is the name of the schema to which Data Integration Hub connects.

Microsoft SQL Server

Example syntax for a Microsoft SQL Server:

```
jdbc:informatica:sqlserver://myhost:1433;databaseName=MY_DB;loginTimeout=30;
```

If you use a named Microsoft SQL Server database instance, specify the instance name in the following format:

```
<server name>\<instance name>
```

where `server name` is the name of the database server and `instance name` is the name of the database instance on which Data Integration Hub is running.

If you run multiple instances on the same database server with unique port numbers, you can specify the instance port number instead of the instance name in the following format:

```
<server name>:<port number>
```

where `server name` is the name of the database server and `port number` is the logical address of the instance on which Data Integration Hub is running.

Note: If you specify an instance name, verify that Microsoft SQL Server browser is running.

ODBC

Example syntax for an ODBC connection:

```
jdbc:database://host:port;databaseName=<databasename>;
```

where `databasename` is the name of the database that you want to connect.

Configuring the JDBC Drivers for an ODBC Connection

To connect Data Integration Hub to the relational databases using an ODBC connection, add the JDBC driver details to the `dx-configuration.properties` file. You must also add the JDBC drivers to the installation directory.

1. On the machine where Data Integration Hub is installed, open the server and the console copies of the `dx-configuration.properties` file in a text editor from the following locations:
   - `<Data Integration Hub installation directory>/DataIntegrationHub/tomcat/shared/classes/`
   - `<Data Integration Hub installation directory>/conf/`
2. In both files, add the driver class of the relational database for which you create an ODBC connection in the following property:

   `dih.custom.connection.jdbc.drivers`

   The following sample property shows the JDBC driver classes for the MySQL database:
   `dih.custom.connection.jdbc.drivers=org.h2.Driver;com.mysql.jdbc.Driver;com.microsoft.sqlserver.jdbc.SQLServerDriver`
3. Save the files.
4. Copy the JDBC drivers that interact with the relational database for which you create an ODBC connection to the following directory:
   `<Data Integration Hub installation directory>\shared\lib`
5. Restart Data Integration Hub.

Relational Database Connection Data Access Properties

A connection includes the following data access properties:

Use This Connection for Data Access

Enable data access to the connection.

Connection String

Database connection string used to retrieve physical data objects from the database or to write physical data objects to the database.

Example syntax for Oracle:

```
oracle.world
```

Example syntax for Microsoft SQL Server:

```
sqlserver@mydatabase
```

For an ODBC connection, enter the ODBC connection string.

Domain Name

Name of the domain in which the database is running.

Server Name

Name of the server to which the relational database connects.

If you use a named database instance, specify the instance name in the following format:

```
<server_name>\<instance_name>
```

If you run multiple instances on the same database server with unique port numbers, you can specify the instance port number instead of the instance name in the following format:

```
<server_name>:<port_number>
```

where `server_name` is the name of the server to which database connects.

Note: If you specify an instance name, verify that Microsoft SQL Server Browser is running.

Packet Size

Microsoft SQL Server database connected with native driver. Maximum packet size used to transfer data. Used to optimize native drivers for Microsoft SQL Server.

Code Page

Character encoding for the database.

Environment SQL

SQL commands that set the database environment each time you connect to the database. The SQL commands run with each database connection.

Default is disabled.

Oracle databases only: If you do not want to use the default schema to which Data Integration Hub connects, append the following string to the connection string:

```
ALTER SESSION SET CURRENT_SCHEMA=<schema name>
```

You must define the same schema in the Metadata Access section, in the **Connection String** field.

Transaction SQL

SQL commands that set the database environment at the beginning of each transaction. The SQL commands run before the initiation of each transaction.

Default is disabled.

Connection Retry Period

Number of seconds to wait before reconnecting to the database in case of connection failure.

Default value is 0.

Driver Type

The type of driver that you use to create a relational database connection. Choose one of the following options:

- **Native**. Data Integration Hub is installed on a Windows operating system.
- **ODBC**. Data Integration Hub is installed on a UNIX or a Linux operating system.

Relational Database Connection Permissions Properties

A relational database connection includes the following permissions properties:

Available Categories

List of categories that you can assign to the connection.

Selected Categories

List of assigned categories for the connection.

When you assign categories to a connection, only user groups with permissions to the connection can perform the following actions:

- View, edit, and delete topics that use the connection to add tables from a database.
- Select topics that use the connection to add tables from a database when creating publications and subscriptions.
- Assign the connection to publications and subscriptions.
- View, edit, run, and delete publications and subscriptions that use the connection.
- View and perform actions on events of publications and subscriptions that use the connection.

If no categories are assigned to the connection, all Data Integration Hub users have permissions to the connection.

Teradata Connection Properties

A Teradata connection can include the following connection details:

- General connection properties, including credentials of the database user account.
- Metadata access connection details. Data Integration Hub uses the metadata access connection details to access the database schema information when the operator creates or edits a topic and adds tables from a database.
- Data access connection details. Data Integration Hub uses the data access connection details at run time to connect to the source application or to the target application.
- Permissions. User categories that have permissions to the connection.

Note: Before you create a Teradata connection, copy the Teradata driver jar files to the Data Integration Hub installation folder. For details, see "Managing Connections" on page 76.

Teradata General Connection Properties

A Teradata connection includes the following general properties:

Connection Name

Name of the connection. The name can contain a maximum of 64 characters. The name cannot contain spaces or other special characters except for the underscore.

Description

Optional description of the connection.

User

Name of the database user account. The database user must have read and write permissions to the database.

Password

Password for the database user account. The password must use 7-bit ASCII encoding.

Teradata Metadata Access Connection Properties

A Teradata connection includes the following metadata access properties:

Use This Connection for Metadata Access

Enable metadata access to the connection.

Connection String

Database connection string used to preview data and create automatic mappings.

Example syntax:

```
jdbc:teradata://host/DBS_PORT=port,DATABASE=dbName;
```

Teradata Data Access Connection Properties

A Teradata connection includes the following data access properties:

Use This Connection for Data Access

Enable data access to the connection.

TDPID

The name of the Teradata database machine.

Database Name

Teradata database name.

Tenacity

Amount of time, in hours, that Data Integration Hub continues trying to log on when the maximum number of operations runs on the Teradata database.

Must be a positive, non zero integer. Default is 4.

Max Sessions

Maximum number of sessions that Data Integration Hub establishes with the Teradata database.

Must be a positive, non zero integer. Default is 4.

Min Sessions

Minimum number of Data Integration Hub sessions required for the Data Integration Hub job to continue.

Must be a positive integer between 1 and the Max Sessions value. Default is 1.

Sleep

Amount of time, in minutes, that Data Integration Hub pauses before it retries to log on when the maximum number of operations runs on the Teradata database.

Must be a positive, non zero integer. Default is 6.

Enable Data Encryption

Enables full security encryption of SQL requests, responses, and data.

Default is disabled.

Block Size

Maximum block size, in bytes, Data Integration Hub uses when it returns data to the PowerCenter Integration Service.

Minimum is 256. Maximum is 64,330. Default is 64,000.

System Operator

Data Integration Hub operator type:

- Export. Exports large data sets from Teradata tables or views. Select Export to use the connection in publications.
- Update. Batch updates, inserts, upserts, and deletes data in Teradata database tables. Select Update to use the connection in subscriptions.

Default is Update.

Code Page

Code page associated with the database.

When you run a session that extracts from a Teradata source, the code page of the Data Integration Hub connection must be the same as the code page of the Teradata source.

Teradata Permissions Connection Properties

A Teradata connection includes the following permissions properties:

Available Categories

List of categories that you can assign to the connection.

Selected Categories

List of assigned categories for the connection.

When you assign categories to a connection, only user groups with permissions to the connection can perform the following actions:

- View, edit, and delete topics that use the connection to add tables from a database.
- Select topics that use the connection to add tables from a database when creating publications and subscriptions.
- Assign the connection to publications and subscriptions.
- View, edit, run, and delete publications and subscriptions that use the connection.
- View and perform actions on events of publications and subscriptions that use the connection.

If no categories are assigned to the connection, all Data Integration Hub users have permissions to the connection.

HDFS Connection Properties

An HDFS connection includes the following connection details:

- General connection properties.
- Hadoop cluster settings.
- Permissions. User categories that have permissions to the connection.

HDFS Connection General Properties

An HDFS connection includes the following general properties:

Connection Name

Name of the connection. The name can contain a maximum of 64 characters. The name cannot contain spaces or other special characters except for the underscore.

Description

Optional description of the connection.

Connection Type

The connection type.

HDFS Connection Hadoop Settings Properties

An HDFS connection includes the following Hadoop settings properties:

Hadoop Cluster User

Name of the Hadoop cluster user account.

NameNode URI

Use the following connection URI:

```
hdfs://<namenode>:<port>
```

Where

- `<namenode>` is the host name or IP address of the NameNode.
- `<port>` is the port on which NameNode listens for remote procedure calls (RPC).

For example:

```
hdfs://mycluster:8020
```

Note: `hdfs://<namenode>:<port>` must be identical to property `fs.defaultFS` as it appears in the file `core-site.xml`.

Hadoop Distribution

Type of Hadoop distribution that the Hadoop cluster uses.

HDFS Connection Permissions Properties

An HDFS connection includes the following permissions properties:

Available Categories

List of categories that you can assign to the connection.

Selected Categories

List of assigned categories for the connection.

When you assign categories to a connection, only user groups with permissions to the connection can perform the following actions:

- View, edit, and delete topics that use the connection to add tables from a database.
- Select topics that use the connection to add tables from a database when creating publications and subscriptions.
- Assign the connection to publications and subscriptions.
- View, edit, run, and delete publications and subscriptions that use the connection.
- View and perform actions on events of publications and subscriptions that use the connection.

If no categories are assigned to the connection, all Data Integration Hub users have permissions to the connection.

File Transfer Connection Properties

A file transfer connection includes the following connection details:

- General connection properties.
- Data access connection details. Data Integration Hub uses the data access connection details at run time to connect to the source application or to the target application.
- Authentication. Enter authentication credentials for the file transfer server. For SFTP connections you can select to authenticate to the SFTP server with a password or to authenticate with a private key.
- Permissions. User categories that have permissions to the connection.

File Transfer General Connection Properties

A file transfer connection includes the following general properties:

Connection Name

Name of the connection. The name can contain a maximum of 64 characters. The name cannot contain spaces or other special characters except for the underscore.

Description

Optional description of the connection.

Connection Type

The connection type.

File Transfer Data Access Connection Properties

A file transfer connection includes the following data access properties:

Host

Host name of the file transfer server.

Port Number

Port number of the file transfer server.

Connection Mode

Applicable for FTP and FTPS connections. Select the connection mode for data connections, passive or active.

SSL Mode

Applicable for FTPS connections. Select SSL mode for data connections, explicit or implicit.

File Transfer Authentication Connection Properties

A file transfer connection includes the following authentication properties:

User

Name of the file transfer server user. The user must have read and write permissions to the server.

Select how to complete authentication

Applicable for SFTP connections. Select the method by which the user you defined authenticates to the file transfer server, and then enter the required information for the method that you select.

Authenticate with password

Enter a password in the **Password** field.

Authenticate with private key from key alias

Select a key alias. Optionally, enter the key passphrase. For more information about the management of key aliases, see Import and Export Utility Use the import and export utility to export objects from the Data Integration Hub repository to an XML file and import the objects from the XML file back to the Data Integration Hub repository. .

Note: This option is only enabled for users with "Use Key Aliases" privileges.

Authenticate with private key from file

Select a key file. Optionally, enter the key passphrase.

Password

Applicable for FTP and FTPS connections and for SFTP connections when the option **Authenticate with password** is selected. Password of the file transfer server user.

File Transfer Permissions Connection Properties

A file transfer connection includes the following permissions properties:

Available Categories

List of categories that you can assign to the connection.

Selected Categories

List of assigned categories for the connection.

When you assign categories to a connection, only user groups with permissions to the connection can perform the following actions:

- View, edit, and delete topics that use the connection to add tables from a database.

- Select topics that use the connection to add tables from a database when creating publications and subscriptions.
- Assign the connection to publications and subscriptions.
- View, edit, run, and delete publications and subscriptions that use the connection.
- View and perform actions on events of publications and subscriptions that use the connection.

If no categories are assigned to the connection, all Data Integration Hub users have permissions to the connection.

Managing Connections

Use the Data Integration Hub Operation Console Navigator to create, edit, test, and delete connections.

Note: Do not manually change or delete connections that you create in Data Integration Hub directly in PowerCenter.

You cannot manage user credentials for the following Data Integration Hub system connections in the Operation Console:

- DIH_REPO. Connection to the Data Integration Hub repository.
- DIH_STAGING. Connection to the Data Integration Hub publication repository.

To manage user credentials for the system connections, see " Managing User Credentials for System Connections" on page 77.

Before you create a Teradata connection, copy the Teradata driver jar files to the following folder:

```
<DIHInstallationDir>\shared\lib\
```

For example:

```
<DIHInstallationDir>\shared\lib\tdgssconfig-14.10.jar
<DIHInstallationDir>\shared\lib\terajdbc4-15.10.jar
```

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

   The **Connections** page appears.
2. Choose to create, edit, or delete a connection. For relational database connections, you can test the connection.
   - To create a connection, click **New Connection**, select the connection type, define the properties for the connection, and then click **Save**.
   - To edit a connection, click the **Edit** icon next to the connection that you want to edit, change properties in the **Edit Connection** page, and then click **Save**.

     Note: You cannot change the type of a connection with associated publications or subscriptions.
   - To test a connection, click the **Test Connection** icon next to the connection that you want to test. Data Integration Hub tests the connection and shows a message with the test results.
   - To delete a connection, click the **Delete** icon next to the connection that you want to delete and confirm the deletion.
     Data Integration Hub deletes the connection in PowerCenter.

     Note: You cannot delete a connection with associated publications or subscriptions.

Related Topics:

- " Managing User Credentials for System Connections" on page 77

Managing User Credentials for System Connections

Manage user credentials for the following Data Integration Hub system connections:

- DIH_REPO. Connection to the Data Integration Hub repository.
- DIH_STAGING. Connection to the Data Integration Hub publication repository.

Note: You cannot manage user credentials for Data Integration Hub system connections in the Data Integration Hub Operation Console.

1. On the machine where Data Integration Hub is installed, open both the server and the console copies of the `dx-configuration.properties` file in a text editor from the following locations:

   ```
   <DIHInstallationDir>/apache-tomcat-version/shared/classes/
   <DIHInstallationDir>/conf/
   ```

2. In both files, change the values of the following properties as required:

Repository	User Name	Password
DIH_REPO	dx.jdbc.username	dx.jdbc.password
DIH_STAGING	dih.staging.jdbc.username	dih.staging.jdbc.password

3. Restart the Data Integration Hub server and the Data Integration Hub Operation Console.

CHAPTER 8

Connectivity to Informatica Intelligent Cloud Services

This chapter includes the following topics:

Connectivity to Informatica Intelligent Cloud Services Overview

Use the Informatica Intelligent Cloud Services Data Integration Hub connector when your organization needs to connect cloud-based applications to Data Integration Hub and you want to use Informatica Intelligent Cloud Services for publication from and subscription to cloud-based applications.

You use the Data Integration Hub connector in Informatica Intelligent Cloud Services to connect to the Data Integration Hub publication repository when you create publication and subscription mappings. You use Data Integration Hub Connector when you create a Data Integration Hub connection, which you then use in Informatica Intelligent Cloud Services mappings and tasks. In mappings and tasks that Data Integration Hub uses for publication, the Data Integration Hub publication repository is the target of the mapping or task. In mappings and tasks that Data Integration Hub uses for subscription, the Data Integration Hub publication repository is the source of the mapping or task.

Connectivity to Informatica Intelligent Cloud Services Administration

Administration of Data Integration Hub connectivity to Informatica Intelligent Cloud Services includes the following tasks:

- Create and maintain an Informatica Intelligent Cloud Services organization. For more information, see the Informatica Intelligent Cloud Services online help.

- Install the cloud Data Integration Hub Connector. For more information, see the Informatica Intelligent Cloud Services online help.
- Configure a cloud Data Integration Hub connection that uses Data Integration Hub Connector. For more information, see the *Cloud Data Integration Hub Connector Guide*.
- Configure cloud connectivity system properties in Data Integration Hub. For more information, see "Cloud Connectivity System Properties" on page 79.

  Note: If you install the Secure Agent on a Linux operating system on which the PowerCenter Integration Service is also installed, install the Secure Agent with a different user than the user that installed the PowerCenter Integration Service.

Connectivity to Multiple Informatica Intelligent Cloud Agents

Administration of Data Integration Hub connectivity to multiple cloud agents on different machines includes the following tasks:

1. Delete the `userparameters` folder within the `<AGENT_HOME>/apps/Data_Integration_Server/data/userparameters` directory on both the different servers where the agent is installed.
2. Create a `userparameters` folder on a common mount, where the folder can be accessed by both the secure agents.
3. Create a softlink from the `/data` directory to the common folder created in step 2.
4. Create a cloud group on Informatica Intelligent Cloud Services and group both the secure agents under the same cloud group.
5. Update the system properties on Data Integration Hub with the below Informatica Intelligent Cloud Services environment variables:
   - dih.ics.installation.folder : You must set this to the install directory of one of the secure agents from one server.
   - dih.ics.runtime.environment : You must set to the Informatica Intelligent Cloud Services cloud group name.
   - dih.ics.url : Set this to https://dm-us.informaticacloud.com
   - dih.ics.username : Set the username.
   - dih.ics.password : Set the password.

Cloud Connectivity System Properties

The following list describes the Data Integration Hub cloud connectivity system properties:

dih.ics.installation.folder

Location where the Informatica Intelligent Cloud Services Secure Agent is installed.

Note: If you install the Secure Agent on a Linux operating system on which the PowerCenter Integration Service is also installed, install the Secure Agent with a different user than the user that installed the PowerCenter Integration Service.

dih.ics.url

URL of the Informatica Intelligent Cloud Services server. The following URL is the default value:

```
https://app.informaticaondemand.com
```

dih.ics.runtime.environment

Runtime environment that contains the Secure Agent that runs automatic cloud publications and subscriptions.

dih.ics.username

User name for the Informatica Intelligent Cloud Services server.

dih.ics.password

Password for the Informatica Intelligent Cloud Services server.

dih.ics.proxy.host

Host name or address of the proxy server to access the Informatica Intelligent Cloud Services server.

dih.ics.proxy.port

Port number of the proxy server to access the Informatica Intelligent Cloud Services server.

dih.ics.proxy.username

User name for the proxy server to access the Informatica Intelligent Cloud Services server.

dih.ics.proxy.password

Password for the proxy server to access the Informatica Intelligent Cloud Services server.

CHAPTER 9

Integration of Data Integration Hub with Enterprise Data Catalog

This chapter includes the following topics:

Integration of Data Integration Hub with Enterprise Data Catalog Overview

The Data Integration Hub administrator integrates Data Integration Hub with Informatica Enterprise Data Catalog to discover and use existing Data Integration Hub objects, and understand their lineage and impact on other entities in the enterprise.

Enterprise Data Catalog brings together all data assets in an enterprise and presents a comprehensive view of data assets and data asset relationships. Data assets in the enterprise might exist in relational databases, purpose-built applications, reporting tools, HDFS, and other big-data repositories.

For example, as an analyst of a major eCommerce retailer, you want to gather data into a Business Intelligence analytics tool. You use Data Integration Hub and the underlying mapping engine to publish data into topics, and configure the Business Intelligence system to subscribe to published topics.

After you configure the PowerCenter mapping and Data Integration Hub resources in Enterprise Data Catalog, you can see the end to end lineage of the analytics business processes in Enterprise Data Catalog.

Data Integration Hub operator can add topic tables from Enterprise Data Catalog assets. For more information about adding topic tables, refer to the *Data Integration Hub Operator Guide*.

A Data Integration Hub scanner retrieves metadata about Data Integration Hub entities such as topics, applications, publications, and subscriptions and builds a relationship diagram.

When you create resources in Enterprise Data Catalog for Data Integration Hub connections and PowerCenter scanner, Enterprise Data Catalog display the flow of data in Data Integration Hub including PowerCenter mappings.

For more information about using Enterprise Data Catalog, refer to the *Enterprise Data Catalog User Guide*.

Configuring Enterprise Data Catalog to Integrate with Data Integration Hub

In order to integrate Data Integration Hub with Informatica Enterprise Data Catalog you must create a Data Integration Hub scanner.

Perform the following steps in the Enterprise Data Catalog Administration Console to configure Data Integration Hub connections:

1. Create a Data Integration Hub scanner with a resource type as Data Integration Hub. The Data Integration Hub scanner makes an API call to retrieve data assets such as publications, subscriptions, topics, and applications.
2. Create resources for Data Integration Hub connections and PowerCenter to view a detailed Data Integration Hub lineage.
3. Run a scan on each of the resources that you created in the Enterprise Data Catalog Administration Console.
4. Link resources by using the following menu option: **Manage** > **Connection Assignment**.

   Note: The Data Integration Hub publication repository connection is named RDBMS_Staging connection in the Enterprise Data Catalog Connection Assignment page.

   For more information about configuring resource connections in the Enterprise Data Catalog Administration Console, see the *Informatica Enterprise Data Catalog Administrator Guide*.

The Data Integration Hub scanner displays Data Integration Hub lineage in Enterprise Data Catalog.

Using Enterprise Data Catalog to View Data Integration Hub Lineage

You can view the lineage and impact for an asset in the **Lineage and Impact** tab in the Enterprise Data Catalog user interface. The lineage diagram displays the flow of data from source to destination.

The Data Integration Hub operator uses the Enterprise Data Catalog user interface to discover and use existing Data Integration Hub objects, and understand their lineage or impact on other entities in the enterprise.
For more information about viewing lineage and impact, see the *Informatica Enterprise Data Catalog User Guide* .

Perform the following steps in the Enterprise Data Catalog user interface to view lineage and impact of Data Integration Hub objects:

1. Navigate to the home page of Enterprise Data Catalog user interface.
2. Search for the name of the Data Integration Hub scanner resource that you created in "Configuring Enterprise Data Catalog to Integrate with Data Integration Hub" on page 82.
3. Click the Data Integration Hub resource that appears in the search results.

   An overview of Data Integration Hub entities displays. Navigate to **Lineage and Impact** and **Relationships** tabs for detailed information.

Topics from Enterprise Data Catalog Assets

The Data Integration Hub operator can add topic tables from Enterprise Data Catalog assets.

To add topic tables from Enterprise Data Catalog assets, Data Integration Hub administrator must configure Enterprise Data Catalog system properties in Data Integration Hub. For more information about configuring Enterprise Data Catalog system properties, see "Enterprise Data Catalog System Properties" on page 59.

For more information about creating topics, refer to the *Data Integration Hub Operator Guide*.

CHAPTER 10

Document Management

This chapter includes the following topics:

Document Management Overview

Data Integration Hub stores files that are published with a publication that is configured with the **When the file is ready to be published** scheduling option in a document store.

When you view an event in the Operation Console for a publication, you can view the document store file that is associated with the event.

Document Store

The document store is a directory where Data Integration Hub stores files that are published with a publication that is configured with the **When the file is ready to be published** scheduling option. The directory must be accessible to the Data Integration Hub server, the Apache Tomcat server, and the PowerCenter Integration Service with the same file path.

You specify the document store directory during Data Integration Hub installation. If you do not specify a directory, the installer uses the following default directory for the document store: `<DIHInstallationDir>\dih-data`

After the installation, Data Integration Hub saves the path to the document store in the document.store.folder system property. You use the repository utility to change the document store directory after installation.

Note: Do not change the value in the system property. If you change the value, you create a file path conflict and Data Integration Hub and PowerCenter cannot access the document store.

Document Store Folder Structure

After you install Data Integration Hub, the Data Integration Hub server creates sub-folders in the document store directory. In addition to the files, the sub-folders store logs and temporary files that Data Integration Hub creates during processing.

The Data Integration Hub creates the following sub-folders in the document store directory:

/tmp

Directory that stores temporary files that Data Integration Hub creates during processing.

Note: Data Integration Hub does not delete temporary files. You must manually clean up the directory periodically. Do not delete temporary files that were created during the last 24 hours before you clean up the directory.

/documents

Directory that stores the files to process. When you pass a reference to PowerCenter, the path must point to files in this directory.

/eventLogs

Directory that stores event logs when you run the Data Integration Hub server in debug mode. Use the event logs for troubleshooting.

Document Store Permissions

You configure permissions for components that need to access the document store. The Operation Console and the PowerCenter Integration Service must have permissions to access the directory.

Do not grant permissions to PowerCenter workflow to write to the /documents sub-folder. PowerCenter workflows must write all files to the /tmp sub-folder.

The following table describes the component permissions to configure:

Component	Permissions
Operation Console	Requires read permission to the following directory: `<DocumentStoreDir>/documents`

PowerCenter Integration Service	- Requires create and write permissions to the following directory:
	`<DocumentStoreDir>/tmp`
	- Requires read permission to the following directory:
	`<DocumentStoreDir>/documents`

Changing the Location of the Document Store

Use the Data Integration Hub repository utility to change the location of the document store directory. The utility changes the value in the system property, updates all the path references in the Data Integration Hub repository, and moves all files to the new directory.

To prevent data loss, perform the following actions before you run the repository utility to change the location of the document store directory:

1. Shut down all Data Integration Hub services.
2. Verify that Data Integration Hub workflows are not running. Verify that there are no JMS messages that contain document references in any Data Integration Hub JMS queues.
3. In the Data Integration Hub repository utility, run the `moveDocumentStore` command with the following syntax in a single line:

```
repoutil –c moveDocumentStore -t dih –l <Data Integration Hub repository jdbc URL> -
u <user name> -p <password> --docStore <new document store location>
```

The following example shows a repoutil script for moving the document store in a node that uses a UNIX operating system:

```
./repoutil.sh -c moveDocumentStore -t dih
-l "jdbc:informatica:oracle://xsvcshacl03:1521;ServiceName=drep02_taf" -u dihadmin -
p mypassword --docStore="/u02/app/infa_shared/DIH_doc_store"
```

For more information about the repository utility, see the section "Data Integration Hub Repository Utility".

Note: Do not move the document store manually. If you manually move the document store, Data Integration Hub will not reference document attachments for events correctly.

CHAPTER 11

Entity Management

This chapter includes the following topics:

Data Integration Hub Entity Management Overview

Data Integration Hub administrator can perform delete operations for multiple entities by using CLIs.

Data Integration Hub operator can delete Data Integration Hub entities such as application, publications, subscriptions, workflows, topics, and connections. If there are multiple entities, then Data Integration Hub administrator can delete Data Integration Hub entities in a bulk instead of deleting entities one after the other.

Deleting Applications

Add names of applications that you must delete from Data Integration Hub in an XML file and use the delete command to delete all applications mentioned in the file.

Syntax

Use the following command syntax to delete multiple applications.

```
deleteentities.bat [-e] Application -f <file> [-p <password>] [-P <password>] [-u <user
name>][--server <hostname:port>]
```

Parameters

The following table describes the command parameters:

Parameter	Description
-e	Include this parameter to define the type of entity that you want to delete.
-f	Absolute path of the file including the name of the file that lists application names.
-u	Include this parameter to enter the user name of Data Integration Hub. User ID must specify the user name separated by the security domain with an @ symbol. For example, `Administrator@informatica.com`
-p	Include this parameter to update the password of the user account of Data Integration Hub
-P	Include this parameter to update the password of the Data Integration Hub operation console.

Example

The following example command deletes applications from Data Integration Hub.

```
deleteentities.bat -e Application -f /data/test/input.xml -u sys -p sys
```

An example of the content in the `input.xml` file is as follows:

```
<applications>
<application>SALESFORCE</application>
<application>NOTEXIST</application>
</applications>
```

Deleting Connections

Enter names of connections that you must delete from Data Integration Hub in an XML file and use the delete command to delete all connections that are mentioned in the file.

Syntax

Use the following command syntax to delete multiple entities.

```
deleteentities.bat [-e] Connection -f <file> [-p <password>] [-P <password>] [-u <user
name>][--server <hostname:port>]
```

Parameters

The following table describes the command parameters:

Parameter	Description
-e	Include this parameter to define the type of entity that you want to delete.
-f	Absolute path of the file including the name of the file that lists application names.

Parameter	Description
-u	Include this parameter to enter the user name of Data Integration Hub. User ID must specify the user name separated by the security domain with an @ symbol. For example, `Administrator@informatica.com`
-p	Include this parameter to update the password of the user account of Data Integration Hub
-P	Include this parameter to update the password of the Data Integration Hub operation console.

Example

The following example command deletes connections from Data Integration Hub.

```
deleteentities.bat -e Connection -f /data/test/input.xml -u sys -p sys
```

An example of the content in the `input.xml` file is as follows:

```
<connections>
<connection>SALESFORCECONNECTION</connection>
<connection>AZURESQLCONNECTION</connection>
<connection>SFTPCONNECTION</connection>
</connections>
```

Deleting Publications

Add names of publications that you must delete from Data Integration Hub in an XML file and use the delete command to delete all publications mentioned in the file.

Syntax

Use the following command syntax to delete multiple publications.

```
deleteentities.bat [-e] Publication -f <file> [-p <password>] [-P <password>] [-u <user
name>][--server <hostname:port>]
```

Parameters

The following table describes the command parameters:

Parameter	Description
-e	Include this parameter to define the type of entity that you want to delete.
-f	Absolute path of the file including the name of the file that lists application names.
-u	Include this parameter to enter the user name of Data Integration Hub. User ID must specify the user name separated by the security domain with an @ symbol. For example, `Administrator@informatica.com`
-p	Include this parameter to update the password of the user account of Data Integration Hub
-P	Include this parameter to update the password of the Data Integration Hub operation console.

Example

The following example command deletes publications from Data Integration Hub.

```
deleteentities.bat -e Publication -f /data/test/input.xml -u sys -p sys
```

An example of the content in the `input.xml` file is as follows:

```
<publications>
<publication>ConnectedCar</publication>
<publication>EmployeePublication</publication>
</publications>
```

Deleting Subscriptions

Add names of subscriptions that you must delete in an XML file and use the delete command to delete all subscriptions mentioned in the file.

Syntax

Use the following command syntax to delete multiple subscriptions.

```
deleteentities.bat [-e] Subscription -f <file> [-p <password>] [-P <password>] [-u <user
name>][--server <hostname:port>]
```

Parameters

The following table describes the command parameters:

Parameter	Description
-e	Include this parameter to define the type of entity that you want to delete.
-f	Absolute path of the file including the name of the file that lists application names.
-u	Include this parameter to enter the user name of Data Integration Hub. User ID must specify the user name separated by the security domain with an @ symbol. For example, `Administrator@informatica.com`
-p	Include this parameter to update the password of the user account of Data Integration Hub
-P	Include this parameter to update the password of the Data Integration Hub operation console.

Example

The following example command deletes subscriptions from Data Integration Hub.

```
deleteentities.bat -e Subscription -f /data/test/input.xml -u sys -p sys
```

An example of the content in the `input.xml` file is as follows:

```
<subscriptions>
<subscription>ConnectedCar</subscription>
<subscription>EmployeeSubscription</subscription>
</subscriptions>
```

Deleting Topics

Add the names of topics that you must delete in an XML file and use the delete command to delete all topics mentioned in the file.

Syntax

Use the following command syntax to delete multiple topics.

```
deleteentities.bat [-e] Topic -f <file> [-p <password>] [-P <password>] [-u <user name>]
[--server <hostname:port>]
```

Parameters

The following table describes the command parameters:

Parameter	Description
-e	Include this parameter to define the type of entity that you want to delete.
-f	Absolute path of the file including the name of the file that lists application names.
-u	Include this parameter to enter the user name of Data Integration Hub. User ID must specify the user name separated by the security domain with an @ symbol. For example, `Administrator@informatica.com`
-p	Include this parameter to update the password of the user account of Data Integration Hub
-P	Include this parameter to update the password of the Data Integration Hub operation console.

Example

The following example command deletes topics from Data Integration Hub.

```
deleteentities.bat -e Topic -f /data/test/input.xml -u sys -p sys
```

An example of the content in the `input.xml` file is as follows:

```
<topics>
<topic>Department</topic>
<topic>Employee</topic>
</topics>
```

Deleting Workflows

Add the names of workflows that you must delete in an XML file and use the delete command to delete all workflows mentioned in the file.

Syntax

Use the following command syntax to delete multiple workflows.

```
delete-entities.sh [-e] Workflow -f <file> [-p <password>] [-P <password>] [-u <user
name>][--server <hostname:port>]
```

Parameters

The following table describes the command parameters:

Parameter	Description
-e	Include this parameter to define the type of entity that you want to delete.
-f	Absolute path of the file including the name of the file that lists application names.
-u	Include this parameter to enter the user name of Data Integration Hub. User ID must specify the user name separated by the security domain with an @ symbol. For example, `Administrator@informatica.com`
-p	Include this parameter to update the password of the user account of Data Integration Hub
-P	Include this parameter to update the password of the Data Integration Hub operation console.

Example

The following example command deletes workflows from Data Integration Hub.

```
deleteentities.bat -e Workflow -f /data/test/input.xml -u sys -p sys
```

An example of the content in the `input.xml` file is as follows:

```
<workflows>
<workflow>WF_DailyDataRecord</workflow>
<workflow>WF_RealtimePublication</workflow>
</workflows>
```

CHAPTER 12

Export and Import

This chapter includes the following topics:

Export and Import Overview

You can export entities from one Data Integration Hub repository and then import the entities into another Data Integration Hub repository.

For example: Export entities from a test environment and import them into the production environment.

You cannot export data from or import data into the publication repository or the operational data store.

You can export and import the following types of entities:

- Applications
- Topics
- Publications
- Subscriptions
- Connections
- Workflows
- Monitoring rules

You can export and import a single entity type, multiple entity types, or all entities types. When you import entities, you can choose the actions to take when entities that you select to import exist in the target repository.

To export or import Data Integration Hub entities, you must log in to Data Integration Hub with a user that belongs to a user group that has the Administrator user role and that meets the following requirements:

- User group permissions are **Grant read and write permissions to all categories**.
- The user group has export data permissions or import data permissions for the entity types to export and import, or both.

- To export and import applications, topics, publications, subscriptions, connections, and workflows, the user group has read privileges or write privileges for the entity types to export and import, or both.

For more information, see "User Groups" on page 49.

You can also use the import and export command line utility to export and import data objects between Data Integration Hub repositories. You can also use the utility to manage private keys that are used for authentication in SSH FTP connections.

Conflict Resolution

When you import entities into a Data Integration Hub repository, you choose the actions to take when entities that you select to import exist in the target repository.

You can choose one of the following resolutions for each entity type:

- Overwrite. Overwrite the entity with the imported entity. Overwritten entities cannot be recovered.
- Reuse. Do not import the entity and keep the existing entity.
- Cancel. Cancel the import operation.
- Skip. Do not import the entity. If the entity exists in the target repository, keep the existing entity.

  Note: Data Integration Hub skips the import of the entity even if the entity does not exist in the target repository.

Exporting Entities

Export Data Integration Hub entities from the Data Integration Hub repository to an export file. You can then import the entities to another Data Integration Hub repository.

Data Integration Hub saves the entities you export to a `gzip` file in a location that you select.

1. In the Navigator, click **Administration** > **Export and Import**.
2. In the **Export** tab, click **Select Entities**.

   The **Select Entities** page appears.
3. From the **Object Type** list, select the type of entities to export or select **All** to export all entity types.

   Entities of the type or types that you select show in the **Available Entities** list.
4. In the **Available Entities** list select the entities to export, click **Add**, and click **OK**. You can select a single entity or multiple entities. To select all the entities in the list, click **Add All**.
5. Click **Export**. In the **Save As** dialog box, define the location and name of the export file and click **Save**.

   Data Integration Hub exports the entities to the export file.

Importing Entities

Import Data Integration Hub entities to the Data Integration Hub repository from a Data Integration Hub export file.

1. In the Navigator, click **Administration** > **Export and Import**.
2. Select the **Import** tab, click **Upload File**, select the export file, and click **OK**.

   The **Entities to Import** area lists the entities in the export file and the action that Data Integration Hub takes for each entity. The action depends on whether or not the entity exists in the target repository and on the conflict resolution of the entity type. The default conflict resolution for all entity types is to overwrite the existing entity with the imported entity.
3. Optionally, change the default conflict resolutions. You can choose one of the following resolutions for each entity type:
   - Overwrite. Overwrite the entity with the imported entity. Overwritten entities cannot be recovered.
   - Reuse. Do not import the entity and keep the existing entity.
   - Cancel. Cancel the import operation.
   - Skip. Do not import the entity. If the entity exists in the target repository, keep the existing entity.

     Note: Data Integration Hub skips the import of the entity even if the entity does not exist in the target repository.

   In the **Entities to Import** area, the descriptions of the actions that Data Integration Hub takes for the entities to import change according to the selected conflict resolution of the entity type.

   Note: If you select a **Cancel** conflict resolution for an entity type, and the export file contains entities of that type, you cannot continue with the import operation.
4. Click **Import**.

   Data Integration Hub imports the selected entities to the repository. If a selected entity exists in the repository, the action that Data Integration Hub takes depends on the conflict resolution of the entity type.

Import and Export Utility

Use the import and export utility to export objects from the Data Integration Hub repository to an XML file and import the objects from the XML file back to the Data Integration Hub repository. You can also use the utility to manage private keys that are used for authentication in SSH FTP connections.

For example:

- Export objects from a test environment and then import them to the production environment.
- Export object metadata for temporary backup.
- Create, update, and delete private keys.

When you export objects with the utility, you create an export specification file to define the objects to export from Data Integration Hub. When you import objects with the utility, you create an import specification file to specify the objects to import and the conflict resolution for duplicate objects.

Consider the following information before you import or export files.

- You cannot import files that were exported from versions of Data Integration Hub that are earlier than Data Integration Hub 10.2.1 into later versions of Data Integration Hub.
- You cannot import and export publications between different database types. For example, you cannot export a publication from a Microsoft SQL server repository and then import the publication into an Oracle server repository.
- To import publications and subscriptions, the structure of the associated topic must be identical to the structure of the topic in the target system. When you import a publication or a subscription with an updated structure, you must apply the overwrite option to both the updated publication or subscription and to the associated topic, so that Data Integration Hub updates the topic structure accordingly.
- To import an HDFS connection, the distribution version must be supported in the current release. You can update the distribution version in the export file before you import the connection.

To run the import and export utility, you must have the following permissions and privileges:

- Read and write permissions for all objects.
- Export privileges to run the export process and import privileges to run the import process.
- To export and manage private keys, you must be logged in as an Administrator and must have the Use Key Aliases privilege.

If you do not have the required permissions and privileges, Data Integration Hub aborts the import or the export process, or the action that you try to perform on private keys.

You can export and import all objects or choose the export/import objects in a specification XML file. The utility exports or imports objects based on the information in the specification file. If you export objects with dependent objects, the export process exports the dependent objects to the export XML file.

Note: Before you import a topic, verify that the storage location of the topic exists on the Data Integration Hub publication repository. On a Microsoft SQL Server database, the file group must exist. On an Oracle database, the table space must exist.

You can find the import and export utility in the following directory:

```
<Data Integration Hub Installation Directory>\dx-tools\
```

The directory also contains the export-all and import-all batch scripts. Use the scripts to export or import all objects. If you use the scripts, you do not need to create a specification file.

You can find sample specification files in the following directory:

```
<Data Integration Hub Installation Directory>\dx-tools\samples
```

Export-All and Import-All Batch Scripts

Use the export-all and import-all batch scripts to export and import all objects in the Data Integration Hub repository. You do not need a specification file to export or import all objects.

Note: By default, the export-all batch script does not export private keys that are associated with SSH FTP connections where the authentication type is **Authenticate with private key from key alias**. To export the private keys you must uncomment the element with the `keyalias` object in the export-all script.

You can find the batch scripts in the following directory:

```
<DIHInstallationDir>\dx-tools\
```

The following list describes the batch scripts.

export-all

Exports all objects from the Data Integration Hub repository to an export XML file. The script sets the import specification file to:

```
samples/export-all-specification-sample.xml
```

import-all

Imports all objects from the export XML file into the Data Integration Hub repository. The script sets the import specification file to:

```
samples/import-all-specification-sample.xml
```

The following list describes parameters and arguments for the export-all and import-all batch scripts.

-f or --file

Argument: ImportFile or ExportFile

Required. Absolute path and file name of the object import file or export file. If you run the import command, this is the file from which to import objects. If you run the export command, this is the file to which to export objects. If you uncomment the element with the `keyalias` object in the export-all script, this is the location of the private keys.

-u or --user

Argument: UserID

Optional. Identifier of the Data Integration Hub user account to use when the import and export utility accesses the Data Integration Hub repository.

The user account must have the following privileges:

- Data Access permissions to all the data in the repository. The user must be a member of a user group for which the permission option **Grant read and write permissions to all categories** is selected.
- Export Data. Required when you export objects.
- Import Data. Required when you import objects.

If you use Informatica domain authentication or Informatica domain with Kerberos authentication, enter the full user ID in the following format:

```
user@domain
```

-U

Argument: Environment variable.

Optional. Environment variable that contains a user name.

User name of an Operation Console user account with Manage Data privileges to run the import or export command. To run the import command, the user account must have the Import Data privilege. To run the export command, the user account must have the Export Data privilege.

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 or --password

Argument: password.

Optional. Password of the Data Integration Hub user account to use when the import and export utility accesses the Data Integration Hub repository.

-P

Argument: Environment variable.

Optional. Environment variable that contains a password.

Password for the Operation Console user that runs the import or the export 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.

--server

Argument: “hostname:port”

Optional. Host name and port number of the Data Integration Hub server. If you do not enter an argument, the import and export utility connects to the localhost server with the default port 18095. You must enclose the argument in quotation marks. For example:

```
--server "localhost:18095"
```

--test

Optional for the import or export process. Not applicable for the management of private keys. Runs the import or export command and generates a report with a list of objects to export or import without exporting or importing the objects. Use this parameter to test the import or export specification file.

The import and export utility creates the report in the same directory as the specification file. The name of the report file is the same as the name of the specification file with a log suffix in the following format:

```
<SpecFileName>.log.xml
```

Repository Objects to Export and Import

You can export or import all objects or specify the export/import object types. When you export or import objects with dependent objects, the import and export utility exports the metadata for the dependent objects. When you create the export or the import specification file, you specify the object type to export or import.

The utility does not export or import event attachments or published data sets. After you import publications or subscriptions, Data Integration Hub creates all associated entities, such as connections and automatic mappings. Imported topics, publications, and subscriptions become valid after the associated entities are created. For connections, the metadata status becomes NOT_TESTED after the import. Connections data access status becomes NOT_TESTES if the PowerCenter connection entity is created successfully and INVALID if creating the PowerCenter connection entity fails..

The following table describes the parent and dependent object types that you can export and import:

Object	Type	Dependent Objects
Application	dihapplication	Topics, publications, subscriptions, connections, workflows, event attributes, categories
Connection	dihconnection	Export: None Import of SSH FTP connections, where the authentication type is **Authenticate with private key from key alias**: key alias.

Object	Type	Dependent Objects
Event Attribute	eventattribute	None
Event Type	eventtype	None
Key Alias	keyalias	None
Publication	dihpublication	Topics, connections, workflows for custom mappings
Subscription	dihsubscription	Topics, connections, workflows for custom mappings
Topic	dihtopic	None
Workflow for a custom mapping	dihworkflow	Event attributes

Export Specification File

The export specification file contains instructions on which objects to export from the Data Integration Hub repository. You use the sample specification file structure to create the import specification file.

When the import and export utility exports an object, it exports all the dependent objects to retain the validity of the exported objects. For example, when you export an application, the utility exports associated publications and subscriptions.

Note: When you export an SSH FTP connection, the import and export utility does not export the associated key alias. You must manually add the key alias to the export specification file.

You can use the (ALL) token to export all object of the same type from the repository. You enter the token instead of the object name.

The following example shows the contents of the export specification file:

```
<ExportSpecification>
    <!-- Export the finance application -->
    <ObjectSelection type="dihapplication">
        <Name>Finance</Name>
    </ObjectSelection>

    <ObjectSelection type="dihapplication">
        <Name>HR</Name>
    </ObjectSelection>

       <!-- Export the finance department topic -->
    <ObjectSelection type="dihtopic">
        <Name>FinanceDepartment</Name>
    </ObjectSelection>

       <!-- Export the employee list application -->
    <ObjectSelection type="dihpublication">
        <Name>EmployeeList</Name>
    </ObjectSelection>

       <!-- Export the HR application -->
    <ObjectSelection type="dihsubscription">
        <Name>HR</Name>
    </ObjectSelection>

       <!-- Export the custom workflow for finance. This includes the finance
publications and the subscriptions that finance subscribed to. -->
    <ObjectSelection type="dihworkflow">
```

```
        <Name>CustomFinanceBackup</Name>
    </ObjectSelection>

    <!--  You must be in an Administrator role and have the Use Key Aliases privilege to
export private keys  -->
    <ObjectSelection type="keyalias">
        <Name>FinanceSourceSshPrivateKey</Name>
    </ObjectSelection>

    <!-- Export the connections for the finance application (explicitly) -->
    <ObjectSelection type="dihconnection">
        <Name>FinanceSourceConnection</Name>
    </ObjectSelection>

    <ObjectSelection type="dihconnection">
        <Name>FinanceTargetConnection</Name>
    </ObjectSelection>

    <!-- Also include any of the event types and event attributes -->
    <ObjectSelection type="eventtype">
        <Name>FinanceEventType_One</Name>
    </ObjectSelection>
    <ObjectSelection type="eventtype">
        <Name>FinanceEventType_Two</Name>
    </ObjectSelection>

    <ObjectSelection type="eventattribute">
        <Name>(ALL)</Name>
    </ObjectSelection>
</ExportSpecification>
```

You must use valid XML names in the export specification file. The following table describes the special characters that you must encode:

Special Character	Encoded Character
<	&lt
>	&gt
&	&amp

Import Specification File

The import specification file contains instructions on which objects to import into the Data Integration Hub repository. You use the sample specification file structure to create the import specification file.

The import specification file defines how to resolve conflicts when an object that you want to import exists in the Data Integration Hub. The import and export utility determines whether an object exists in the Data Integration Hub based on the object name.

The following table describes the conflict resolution types you can set:

Conflict Resolution Type	Description
cancel	Cancels the import process. No objects are imported.
overwrite	Overwrites the object in the Data Integration Hub repository with the object from the export file. If you choose to overwrite existing objects, the utility performs partial validation during the import process. For example, the utility does not verify whether the user objects have associated categories. Information about missing objects appears in the import log file. If you choose to overwrite objects that do not exist, the utility creates the objects.
reuse	Retains the object in the Data Integration Hub repository if the object name matches the object in the export XML file, and does not import of the object from the export file.
skip	Skips the object in the export XML file and does not check whether the object exists in the Data Integration Hub repository.
default	Applies the conflict resolution that you set at the parent level of the specification file. For example, if you set the object type resolution to **reuse**, you can create an element for a specific object name and set the resolution type to **default**. The import and export utility applies the object type resolution to the specific object.

Conflict resolution types are case sensitive.

You can specify the conflict resolution for all objects, for a type of object, or for a specific object. The following elements define the scope of the resolution that you can perform:

- DefaultResolution. Applies the conflict resolution you set to all objects to import. You can define set the default resolution to any type except for default.
- ObjectTypeResolutions. Applies the conflict resolution you set to objects of the same type. Use the type attribute to specify the object types.
- ObjectResolution. Applies the conflict resolution you set to specific object names. Use the name element to specify the object names.

The following example shows the contents of an import specification file that defines the conflict resolution for objects and object types:

```
<ImportSpecification>
    <!-- If an object already exists then reuse it -->
     <DefaultResolution>reuse</DefaultResolution>

    <ObjectTypeResolutions>
     <!-- Cancel the entire import whenever a conflicting connection is found in the
target system -->
        <ObjectTypeResolution type="dihconnection">cancel</ObjectTypeResolution>
        <!-- Reuse any existing application from the target system -->
        <ObjectTypeResolution type="dihapplication">reuse</ObjectTypeResolution>
    </ObjectTypeResolutions>
    <ObjectTypeResolutions>
     <!-- Skip over the connection, these are already defined in the target system-->
        <ObjectTypeResolution type="dihconnection">skip</ObjectTypeResolution>
  <!-- Cancel the entire import whenever a conflicting workflow is found in the target
system -->
        <ObjectTypeResolution type="dihworkflow">cancel</ObjectTypeResolution>
  <!--Overwrite any publication-->
     <ObjectTypeResolution type="dihpublication">overwrite</ObjectTypeResolution>
    </ObjectTypeResolutions>
```

```
  <!--Overwrite the specific application 'Finance'-->
     <ObjectResolution type="dihapplication">
        <Name>Finance</Name>
        <Resolution>overwrite</Resolution>
     </ObjectResolution>

  <!--Skip the subscription 'HR' during import-->
     <ObjectResolution type="dihsubscription">
        <Name>HR</Name>
        <Resolution>skip</Resolution>
     </ObjectResolution>

<!--Reuse the topic 'FinanceDepartment' during import-->
     <ObjectResolution type="dihtopic">
        <Name>FinanceDepartment</Name>
        <Resolution>reuse</Resolution>
     </ObjectResolution>
</ImportSpecification>
```

You must use valid XML names in the export specification file. The following table describes the special characters that you must encode:

Special Character	Encoded Character
<	&lt
>	&gt
&	&amp

Import and Export Utility Command Syntax

Use the import and export utility command syntax to define the scope, location, and permissions for the import or export process and for the management of private keys that are used for authentication in SSH FTP connections.

The import and export utility uses a different syntax for the import or export process and for the management of private keys.

Import and export command syntax

```
importexport
<-c|--command> command
<-f|--file> exportfile
<-s|--specification> specfile
<-u|--user> userID
<-p|--password> user password
[--server "hostname:port"]
[--test]
```

For example:

```
importexport import -f "C:\Users\Administrator\DIH_backup\exported_entities.xml" -s
C:/dx-tools/samples/import-all-specification-sample.xml -u Administrator -p
Administrator
```

Private key management command syntax

```
importexport
<-c|--command> command
<-f|--file> exportfile
<-p|--password> user password
<-u|--user> userID
[--server "hostname:port"]
[--action] key alias action
[--alias] alias of the private key
```

For example:

```
importexport managekeys -f "C:\Users\Administrator\Private_keys\PrivateKey.store" -u
Administrator -p Administrator –action create –alias root
```

The following list describes parameters and arguments for the import and export utility commands.

-c or --command

Argument: command

The command to run. Specify one of the following commands:

- export. Exports specific objects from the Data Integration Hub repository based on the export specification file you create. The utility saves the exported objects in an XML file that you can import back into the Data Integration Hub repository.
- import. Imports specific objects from the export XML file into the Data Integration Hub repository based on the import specification file you create.
- managekeys. Manages private keys. You can create, update, and delete private keys.

-f or --file

Argument: ImportFile or ExportFile

Required. Absolute path and file name of the object import file or export file. If you run the import command, this is the file from which to import objects. If you run the export command, this is the file to which to export objects. If you run the managekeys command, this is the location of the private keys.

-s or --specification

Argument: SpecFile

Required for the import or export process. Not applicable for the management of private keys. Absolute path and file name of the import or export specification file.

-u or --user

Argument: UserID

Optional. Identifier of the Data Integration Hub user account to use when the import and export utility accesses the Data Integration Hub repository.

The user account must have the following privileges:

- Data Access permissions to all the data in the repository. The user must be a member of a user group for which the permission option **Grant read and write permissions to all categories** is selected.
- Export Data. Required when you export objects.
- Import Data. Required when you import objects.

If you use Informatica domain authentication or Informatica domain with Kerberos authentication, enter the full user ID in the following format:

```
user@domain
```

-U

Argument: Environment variable.

Optional. Environment variable that contains a user name.

User name of an Operation Console user account with Manage Data privileges to run the import or export command. To run the import command, the user account must have the Import Data privilege. To run the export command, the user account must have the Export Data privilege.

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 or --password

Argument: password.

Optional. Password of the Data Integration Hub user account to use when the import and export utility accesses the Data Integration Hub repository.

-P

Argument: Environment variable.

Optional. Environment variable that contains a password.

Password for the Operation Console user that runs the import or the export 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.

--server

Argument: “hostname:port”

Optional. Host name and port number of the Data Integration Hub server. If you do not enter an argument, the import and export utility connects to the localhost server with the default port 18095. You must enclose the argument in quotation marks. For example:

```
--server “localhost:18095”
```

--test

Optional for the import or export process. Not applicable for the management of private keys. Runs the import or export command and generates a report with a list of objects to export or import without exporting or importing the objects. Use this parameter to test the import or export specification file.

The import and export utility creates the report in the same directory as the specification file. The name of the report file is the same as the name of the specification file with a log suffix in the following format:

```
<SpecFileName>.log.xml
```

--action

Arguments: create, update, or delete.

Optional for the management of private keys. Not applicable for the import or export process. Manage key action.

--alias

Argument: Alias of the private key.

Optional for the management of private keys. Not applicable for the import or export process. Manage key action.

Exporting Objects from the Data Integration Hub Repository

Use the import and export utility to export specific objects from the Data Integration Hub repository to an export XML file. You create an export specification file and specify the object types or names to export.

Note: You can also use the export-all batch script to export all objects. You do not need to create a specification file to export all objects.

1. Create the export specification XML file based on the sample specification file and set the conflict resolutions.

   You can edit the file in a text editor.
2. Save the export specification XML file in a shared directory accessible to the Data Integration Hub server.
3. From the command line, run the **export** command of the import and export utility with the required parameters.

   The utility creates the following files:

   - Object XML file. Contains the metadata of the exported objects. This file also contains the metadata of dependent objects.
   - Log XML file. Contains the export process steps and information about missing objects.

Do not edit the object XML file. If you change, add, or remove content from the file, the import and export utility might not import the objects correctly.

Importing Objects into the Data Integration Hub Repository

Use the import and export utility to import specific objects into the Data Integration Hub repository from the export XML file. You create an import specification file and set the conflict resolution to perform if an object from the export XML file already exists in the Data Integration Hub repository.

Note: You can also use the import-all batch script to import all objects. You do not need to create a specification file to import all objects.

1. Verify that the export XML file is stored in a shared directory accessible by the Data Integration Hub server.
2. Create the import specification XML file based on the sample specification file and set the conflict resolutions.

   You can edit the file in a text editor.
3. Save the import specification XML file in a shared directory accessible to the Data Integration Hub server.
4. From the command line, run the import command of the import and export utility with the required parameters.

   The utility creates an import log file that contains the import process steps and information about missing objects.

CHAPTER 13

Data Integration Hub Utilities

This chapter includes the following topics:

Data Integration Hub Utilities Overview

The Data Integration Hub utilities perform administrative tasks for Data Integration Hub from the Windows or UNIX command line. The utilities include different commands on Windows and UNIX operating systems.

Data Integration Hub includes the following utilities:

Data Integration Hub services

Starts and stops all Data Integration Hub services. The utility is available on Windows operating systems.

The utility is in the following location: `<DIHInstallationDir>/bin/dihservices`

Data Integration Hub console

Starts and stops the Operation Console service. The utility is available on Windows and UNIX operating systems.

The utility is in the following location: `<DIHInstallationDir>/bin/dihconsole`

Data Integration Hub server

Starts and stops the Data Integration Hub server. The utility is available on Windows and UNIX operating systems.

The utility is in the following location: `<DIHInstallationDir>/bin/dihserver`

Data Integration Hub Repository Utility

You can use the repository command line utility to perform maintenance actions on the Data Integration Hub repository, publication repository, and operational data store, such as creating repository tables or deleting content.

Data Integration Hub Services Utility

The Data Integration Hub services utility starts, stops, and manages the registration of all Data Integration Hub services. The utility is available only on Windows operating systems.

You can also start and stop all Data Integration Hub services from the Start menu.

Command Syntax

The Data Integration Hub services utility uses the following syntax:

```
dihservices
<start|stop|install|remove>
```

The following table describes the Data Integration Hub services utility commands:

Command	Description
start	Starts all Data Integration Hub services.
stop	Stops all Data Integration Hub services.
install	Registers all Data Integration Hub services to the Windows registry with default settings.
remove	Removes all Data Integration Hub services from the Windows registry.

Data Integration Hub Console Utility

The Data Integration Hub console utility starts and stops the Apache Tomcat server.

Data Integration Hub uses the Apache Tomcat server to send commands between the Operation Console Web client and the Data Integration Hub server.

On Windows operating systems, you can start the Apache Tomcat server as an application or as a Windows service.

The Apache Tomcat server creates temporary files in the following directory:

```
<DIHInstallationDir\DataIntegrationHub\tomcat
```

Data Integration Hub does not delete the temporary files. You must manually clean up the directory periodically. Do not delete temporary files that were created during the last 24 hours before you clean up the directory.

Windows Command Syntax

On Windows operating systems, the Data Integration Hub console utility uses the following syntax:

```
dihconsole
<start|stop|install|svcstart|svcstop|remove>
```

The following table describes the Data Integration Hub console utility commands:

Command	Description
start	Starts the Apache Tomcat server as an application.
stop	Stops the Apache Tomcat server.
install	Registers the Apache Tomcat server to the registry as a Windows service.
svcstart	Starts the Apache Tomcat server Windows service.
svcstop	Stops the Apache Tomcat server Windows service.
remove	Removes the Apache Tomcat server Windows service from the registry.

UNIX Command Syntax

On UNIX operating systems, the Data Integration Hub console utility uses the following syntax:

```
dihconsole.sh
<start|stop>
```

The following table describes the Data Integration Hub console utility commands:

Command	Description
start	Starts the Apache Tomcat server.
stop	Stops the Apache Tomcat server.

Data Integration Hub Server Utility

The Data Integration Hub server utility starts, stops, and manages the Data Integration Hub server service.

The Data Integration Hub server is the main component that manages data processing in Data Integration Hub.

When you run the Data Integration Hub server utility, you can specify the host name and port number of the Data Integration Hub server. If you do not specify the host name and port number, the utility uses the local host and default port that you specify during installation.

On Windows operating systems, you can start the Data Integration Hub as an application or as a Windows service.

By default, the Data Integration Hub server creates temporary files in one of the following directories:

```
Windows: <SystemDrive>\temp
UNIX: /tmp or /var/tmp
```

Data Integration Hub does not delete the temporary files. You must manually clean up the directory periodically. Do not delete temporary files that were created during the last 24 hours before you clean up the directory.

Windows Command Syntax

On Windows operating systems, the Data Integration Hub server utility uses the following syntax:

```
dihserver
<start [port]|
 stop [host][port]|
 install|
 svcstart|
 svcstop|
 remove|
 console|
 ping [host][port]|
 status>
```

The following table describes the Data Integration Hub server utility commands:

Command	Description
start	Starts the Data Integration Hub server. You can use the default port number or specify a port number.
stop	Stops the Data Integration Hub server. You can use the default host name and port number or specify a host name and port number.
install	Registers the Data Integration Hub server as a Windows service to the registry.
svcstart	Starts the Data Integration Hub server Windows service.
svcstop	Stops the Data Integration Hub server Windows service.
remove	Removes the Data Integration Hub server Windows service from the Windows registry.
console	Starts the Data Integration Hub server as an application. This command is the same as the Start menu option.
ping	Pings the Data Integration Hub server. You can use the default host name and port number or specify a host name and port number.
status	Returns the status of the Data Integration Hub server Windows service. Returns one of the following values: - Not installed - Starting - Started - Stopping - Stopped The following example shows the output message that the status command can return: `The Data Integration Hub server service is not installed.`

UNIX Command Syntax

On UNIX operating systems, the Data Integration Hub server utility uses the following syntax:

```
dihserver.sh
<start [port]|
 stop [host][port]|
 ping [host][port]|>
```

The following table describes the Data Integration Hub server utility commands:

Command	Description
start	Starts the Data Integration Hub server. You can use the default port number or specify a port number.
stop	Stops the Data Integration Hub server. You can use the default host name and port number or specify a host name and port number.
ping	Pings the Data Integration Hub server. You can use the default host name and port number or specify a host name and port number.

Data Integration Hub Repository Utility

Use the Data Integration Hub repository utility to create, upgrade, and maintain the Data Integration Hub repository, the Data Integration Hub publication repository, and the operational data store.

The Data Integration Hub repository contains information about objects and events. The publication repository contains published data sets. The operational data store stores aggregated data about publication events and subscription events.

For example, you can use the repository utility to create a repository if you did not create the repository during installation. You also use the repository utility to change the document store directory or the user authentication mode.

The repository utility is a command-line utility. You can find the utility in the following directory:

```
<Data Integration Hub Installation Directory>\dx-tools\
```

Note: Before you run the repository utility you must stop the Data Integration Hub Operation Console service and the Data Integration Hub server service. After you run the repository, restart the services.

Repository Utility Command Syntax

Use the repository utility syntax to define the actions that you want to perform on the Data Integration Hub repository, on the publication repository, or on the operational data store.

The repository utility uses the following syntax:

```
repoutil
<-c|--command> command
<-t|contentType> contentType
<-l|--url> "url"
<-u|--user> user
<-p|--password> password
[--authmode mode]
[--sysadmin name]
[--docStore docStore]
[--file file]
[--forceDelete]
[-Ddx.kerberos.initial.administrator]
[-Ddx.pwc.domain.gateway]
[-Ddx.kerberos.krb5.file]
[-Ddx.kerberos.console.keytab.file]
[-Ddx.kerberos.console.service.principal.name]
```

The following table describes the repository utility options and arguments:

Options	Argument	Description
- c --command	command	Required. Command to run on the repository. Enter one of the following commands: - createContent. Initializes the repository. Use this command after you run the repository utility with the deleteContent command. - createSchema. Creates the tables and views in the repository. - deleteContent. Removes all content from the repository. If you use this command, you must run the repository utility with the createContent command before you can use the repository again. - deleteSchema. Deletes all tables and views in the repository. - disablePartitioning. Disables partitioning on the publication repository. The partitions state appears in the dih.staging.use.partitions.default system property. - enablePartitioning. Enables partitioning on the publication repository. Data Integration Hub creates a partitioning schema and a partitioning function, and manages all the tables in the publication repository within partitions. You must provide the access credentials of the Data Integration Hub repository in the command, and not the credentials of the Data Integration Hub publication repository. For example: `repoutil –c enablePartitioning –u DIH_USR –p DIH_USR –l ”jdbc:informatica:oracle://dih_rdbms:1521;SID=orcl” –t dih` The partitions state appears in the dih.staging.use.partitions.default system property. **Note:** When you enable or disable partitioning on the publication repository, Data Integration Hub re-generates the publication tables and erases all the publication data. The publications are no longer valid, and you must update them manually for validation. - loadProperties. Loads and sets Data Integration Hub system properties and event attributes in the Data Integration Hub repository. - migrateToISP. Switches the authentication mode to Informatica domain authentication and deletes all user information from the Data Integration Hub repository. For details see the section "Switching to Informatica Domain Authentication". You can synchronize Data Integration Hub with the Informatica security domain when you switch authentication modes. Create a `.PROPERTIES` file with the properties dx.authentication.groups, dx.pwc.domain.gateway, dx.pwc.user, and

Options	Argument	Description
		dx.pwc.password, and enter the file path in the `--file` command to load the properties from the file. You can also synchronize the users after you switch authentication modes on the **Users** page of the Operation Console. - migrateToISPKerberos. Switches the authentication mode to Informatica domain with Kerberos authentication. Deletes synchronized users from the Data Integration Hub repository and adds the system administrator that is defined by the command argument `–Ddx.kerberos.initial.administrator`. For details see the section "Switching to Informatica Domain with Kerberos Authentication". - migrateToNative. Switches the authentication mode to Data Integration Hub native authentication and deletes synchronized users from the Data Integration Hub repository. For details see the section "Switching to Native Authentication". - moveDocumentStore. Moves the document store directory to a different location. Enter the absolute file path of the location to which you want to move the document store in the `--file` command. - upgradeSchema. Upgrades the repository to the latest version. - verifyContents. Verifies the contents of the Data Integration Hub repository.
-t --contentType	contentType	Required. Specifies on which repository the command runs. Specify one of the following options: - dih. Data Integration Hub repository. - dih_staging. Data Integration Hub publication repository. - dih_ods. Operational data store.
-l -- url	"url"	Optional. JDBC URL for the Data Integration Hub repository or for the Data Integration Hub publication repository. You must enclose the URL in quotation marks. For example: `repoutil -c createContent -l "jdbc:informatica:oracle://oracle_1:1521;SID=orcl"...`
-u --user	user	Optional. User name for the database account to use when the utility connects to the Data Integration Hub repository or for the operational data store.
-p --password	password	Optional. Password for the database account to use when the utility connects to the Data Integration Hub repository or for the operational data store.

Options	Argument	Description
--authMode	mode	Optional for the createContent command. The argument determines the authentication mode to set. Specify one of the following options: - native. Native authentication. - isp. Informatica platform authentication.
--sysadmin	name	Required for the following commands: - migrateToNative - createContent if the --authMode parameter value is native Creates an administrator user account that you use to log in to the Operation Console. By default, the password for the administrator user account is the same as the user name. If --authMode has the value "isp", this option is ignored.
--docStore	docStore	Required for the moveDocumentStore command. Absolute path of the directory to which to move the document store. The directory must have the same access permissions as the current directory. You cannot move the document store to a subdirectory of the current document store directory. For example, if current document store directory is `c:\DocStore`, you cannot move the document store to the following directory: `c:\DocStore\newstore` If the repository utility fails when you run the moveDocumentStore command, you can resume the move with the same value in the --docStore command. On Windows operating systems, you must use forward double slashes (//) in the file path.

--file	file	Optional for the loadProperties, migrateToISPKerberos, migrateToISP, and verifyContents commands.
		Enter the following file name in the command:
		`<DIH_InstallationDir>/conf/dx-configuration.properties`
		The commands use the file to determine the connection properties of the publication repository. Each property must appear on a separate line in the following format:
		`<propertyName>=<propertyValue>`

--configFile	propertyFile	Optional for all the repository commands. Points to the location of the Data Integration Hub configuration property file. If not specified, Data Integration Hub loads the file from the following location: `<DIHInstallationDir>/conf/dx-configuration.properties`

Options	Argument	Description
--createVersion	version	Optional. If you run the createSchema command, you can use this option to specify the version of the product for which to create the repository. The product version consists of numbers separated by periods. For example: 9.6.0. By default, the product version is set to the latest version.
--forceDelete		Optional for the deleteSchema command. Deletes the repository schema regardless of errors. By default, the deleteSchema command does not delete the repository schema if it encounters errors.

-Ddx.kerberos.initial.administrator	user	Required for the migrateToISPKerberos command. Kerberos user that exists in the Informatica security domain, in the following format:
		`<username>@<SECURITY_DOMAIN>`
		You must enter `<SECURITY_DOMAIN>` in uppercase letters. For example:
		`Administrator@DEVELOPMENT.COM`

-Ddx.pwc.domain.gateway	host	Required for the migrateToISPKerberos command. Gateway machine to the Informatica domain. For example: `host:6005`
-Ddx.kerberos.krb5.file	file	Required for the migrateToISPKerberos command. Location of the Kerberos configuration file. This file usually resided in the same location as the PowerCenter configuration file.
-Ddx.kerberos.console.keytab.file	file	Required for the migrateToISPKerberos command. Location of the keytab file. If Data Integration Hub is installed on the same machine as the PowerCenter Administrator Console, the keytab file is the same file that is used for the service principal `HTTP/<hostname>@<domain>`, the file `webapp_http.keytab`. If Data Integration Hub is installed on a different machine than the PowerCenter Administrator Console, the keytab file must contain the credentials for the service principal `HTTP/<DIHhostname>@<domain>`.

-Ddx.kerberos.console.service.principal.name	name	Required for the migrateToISPKerberos command. Service principal host name in the following format:
		`HTTP/<FQDN>@<REALM NAME>`
		For exampe:
		`HTTP/ webserver.development.com@DEVELOPMENT.COM`

Repository Utility Command Examples

Create a new Data Integration Hub repository:

```
repoutil -c createSchema -u DIH_USR -p DIH_USR -l "jdbc:informatica:oracle://
dih_rdbms:1521;SID=orcl"  -t dih

repoutil -c createContent -u DIH_USR -p DIH_USR -l "jdbc:informatica:oracle://
dih_rdbms:1521;SID=orcl"  -t dih
```

Create a new Data Integration Hub publication repository:

```
repoutil -c createSchema -u DIH_STAGING_USR -p DIH_STAGING_USR -
l "jdbc:informatica:oracle://dih_rdbms:1521;SID=orcl"  -t dih_staging

repoutil -c createContent -u DIH_STAGING_USR -p DIH_STAGING_USR -
l "jdbc:informatica:oracle://dih_rdbms:1521;SID=orcl"  -t dih_staging
```

Create a new operational data store:

```
repoutil -c createSchema -u DIH_USR -p DIH_ODS_PASSWORD -l "jdbc:informatica:oracle://
dih_rdbms:1521;SID=orcl"  -t dih_ods

repoutil -c createContent -u DIH_USR -p DIH_ODS_PASSWORD -l "jdbc:informatica:oracle://
dih_rdbms:1521;SID=orcl"  -t dih_ods
```

CHAPTER 14

Dashboard and Reports Management

This chapter includes the following topics:

Dashboard and Reports Management Overview

The Dashboard displays personalized visual reports about information that Data Integration Hub processes. Use the Dashboard to view summary information about Data Integration Hub event processing, such as the number of publication events or the number of errors by application.

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. All reports on the Dashboard appear in the Events tab. The reports on the Dashboard using the metadata directory are based on information that Data Integration Hub collects from the metadata directory. 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. KPIs provide measurable information about events that Data Integration Hub processes. The operational data store is a repository that contains aggregated information solely for reporting purposes.

The Dashboard displays the aggregated event information in panels that you view in the **Dashboard** page of the Operation Console. To use the Dashboard using operational data store, you must install and configure the operational data store when you install Data Integration Hub.

The following table describes the default KPIs that the Dashboard uses for reports:

KPI	Description
Publication or subscription processing time	Duration in seconds of the time it takes for the publication or subscription to reach a final state.
Number of events	Number of publication and subscription events that Data Integration Hub processes.
Number of error events	Number of error events that reached a final state.

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. You import the workflow to PowerCenter after you install the Data Integration Hub Dashboard and Reports component. The workflow runs at scheduled intervals, and you can change certain aspects of the workflow behavior. For example, you can configure 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.

In Dashboard and reports system properties, you change certain aspects of the Dashboard behavior. For example, you can choose to show the Dashboard using operational data store when users log in to the Operation Console.

In the Dashboard using operational data store, you can organize the display of the charts, and add tabs to display selected reports. Each report displays event information from the operational data store or the run-time Data Integration Hub repository based on filters that you apply. You can drill to display the events from each panel in the **Event List** page. For more information about specific Dashboard reports, see the *Data Integration Hub Operator Guide*.

Dashboard and Reports System Properties

You can use system properties to modify certain aspects of the Dashboard behavior .

The following table describes the Dashboard and reports system properties:

System Property	Description

dx.dashboard.url	Connection string to the dashboard server, in the following format:
	`http://<hostname>:<port>/<dashboard name>`
	If you use HTTPS to connect to the Operation Console, the URL must match the value of the property. Otherwise, the Dashboard does not appear. For example:
	`https://myhost:18443/dih-dashboard`

dx.dashboard.ods.page.show	Determines whether to show the Dashboard using operational data store on the Operation Console. Default is false. **Note:** To display the reports in the Dashboard using operational data store, you must install the Data Integration Hub Dashboard and Reports component using the installer and set this property to `true` after installation.

System Property	Description
dx.dashboard.max.timewindow	Maximum time frame in hours that Operation Console users can select to display unresolved error events in the Dashboard. Default is 96.
dx.dashboard.jdbc.username	User name for the operational data store database.
dx.dashboard.jdbc.password	Password for the operational data store in an encrypted string database. If you change the password you must encrypt the string with the password encryption utility and use the encrypted string.
dx.dashboard.jdbc.url	Location of operational data store. The location must be different from the Data Integration Hub repository.
dx.dashboard.errorrate.threshold.low	For internal use only. Do not change the default value.
dx.dashboard.errorrate.threshold.high	For internal use only. Do not change the default value.
dx.dashboard.sla.detection.midnight.latency	For internal use only. Do not change the default value.
dx.first.day.of.week	For internal use only. Do not change the default value.
dx.ods.latency.seconds	Number of seconds between the time the event finished processing and the time that the event load process starts. The operational data store event loader loads events for which the processing time difference in seconds is equal or greater than this value. For example, if you increase the latency to 60 seconds, the event loader only loads events that finished processing at least 60 seconds before the load process starts. Default is 0.
dx.ods.row.limit.thousands	Number of thousands of events to load in each batch when the total number of events is higher than this value. If the total number of events to load is less than the value in this property, the operational data store event loader runs one batch. If you set the row limit to 0, the event loader runs one batch regardless of the number of events. Default is 500. Must be numeric and greater than or equal to 0.

Operational Data Store Event Loader

This section applies only if you use the Dashboard using operational data store.The operational data store loader is a PowerCenter workflow that collects event information from the run-time Data Integration Hub repository and then loads the aggregated events to the operational data store. The Dashboard using operational data store retrieves the aggregated event information and displays it in panels based on the selected KPI.

You can change workflow parameters that affect the workflow behavior. For example, you can choose how long to wait between each event load process and how many retry attempts to perform before failing the workflow. Do not change any internal workflow parameters.

The workflow determines which events to load based on the difference between the time that the event finished processing and the time that the scheduled load process starts. Use the dx.ods.latency.seconds

system property to determine the time to wait before the workflow loads the event after the time the event finished processing. Increase the latency if you experience clock sync issues or if you expect events with longer processing time.

If you process a large volume of events, you can change Data Integration Hub system properties to minimize bottlenecks and to increase performance during the event load process. The workflow loads events in batches. Use the dx.ods.row.limit.thousands system property to determine the number of events to include in each batch.

You import the operational data store event loader to PowerCenter after you install the Data Integration Hub Dashboard and Reports component with the main Data Integration Hub installation. For more information, see the *Data Integration Hub Installation and Configuration Guide*.

Note: If a PowerCenter session fails, the operational data store event workflow might not display a failed status. Monitor the PowerCenter session to verify the success of the run.

Operational Data Store Event Loader Configuration

This section applies only if you use the Dashboard using operational data store.Configure the operational data store event loader variables and parameters to modify certain aspects of the workflow behavior. You can only modify the variables and parameters listed in this section. All other variables and parameters are for internal use.

The following table describes the DX_ETL and DX_ETL_SQLSERVER workflow variables and parameters that you can modify:

How to Access	Property	Description
Workflow menu > **Edit** > **Variables tab**	$$WF_Last_Load_End_Time	Last date and time until which the event loader loaded the events to the operational data store. The event loader uses this time as the starting point the next time it loads the events. Default is `01/01/2005 00:00:00.00` **Note:** Do not modify the variable after the workflow runs. You can modify the variable before the first run.
Workflow menu > **Edit** > **Variables tab**	$$WF_Number_Of_Retry_Attempts	Number of times that the event loader attempts to load the events to the operational data store. If the event loader cannot complete successfully after the defined number of attempts, the workflow fails and creates a Data Integration Hub error event. Default is `3`.
Workflow menu > **Edit** > **Variables tab**	$$WF_Wait_Before_Next_Load_Minutes	Number of minutes to wait before the event loader loads the event to the operational data store. Default is `15`.
WAIT_BEFORE_RETRY task > **Edit** > **Timer tab**	Start_After	Number of minutes to wait before a retry attempt in case of an error in the workflow. Default is 1.

Dashboard and Reports Management Rules and Guidelines

When you work with the Dashboard and the operational data store, consider the following rules and guidelines:

- By default, the session timeout for the Dashboard using operational data store is 30 minutes. To ensure consistency with the Operation Console, if you change the session timeout for the Operation Console, change the value of the Dashboard session timeout property in the following file: `<DIHInstallationDir>/tomcat/dih-dashboard/WEB_INF/web.xml`.

INDEX

F

H

I

K

M

N

O

P

R

S

T

U