Informatica

Informatica® Big Data Management
10.2

Big Data Management Administrator Guide

Informatica Big Data Management Big Data Management Administrator Guide
10.2
July 2018

Publication Date: 2018-12-11

Table of Contents

Preface

The *Big Data Management™ Administrator Guide* is written for Informatica administrators. The guide contains information that you need to administer the integration of the Informatica domain with the Hadoop cluster. It includes information about security, connections, and cluster configurations. This guide assumes that you are familiar with the Informatica domain and the Hadoop environment.

Informatica Resources

Informatica Network

Informatica Network hosts Informatica Global Customer Support, the Informatica Knowledge Base, and other product resources. To access Informatica Network, visit https://network.informatica.com.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- 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 search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit https://kb.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

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

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

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access 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. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions.

If you are an Informatica Network member, you can access 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 augment, extend, or enhance your Informatica implementations. By leveraging any of the hundreds of solutions from Informatica developers and partners, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

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

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
http://www.informatica.com/us/services-and-training/support-services/global-support-centers.

If you are an Informatica Network member, you can use Online Support at http://network.informatica.com.

CHAPTER 1

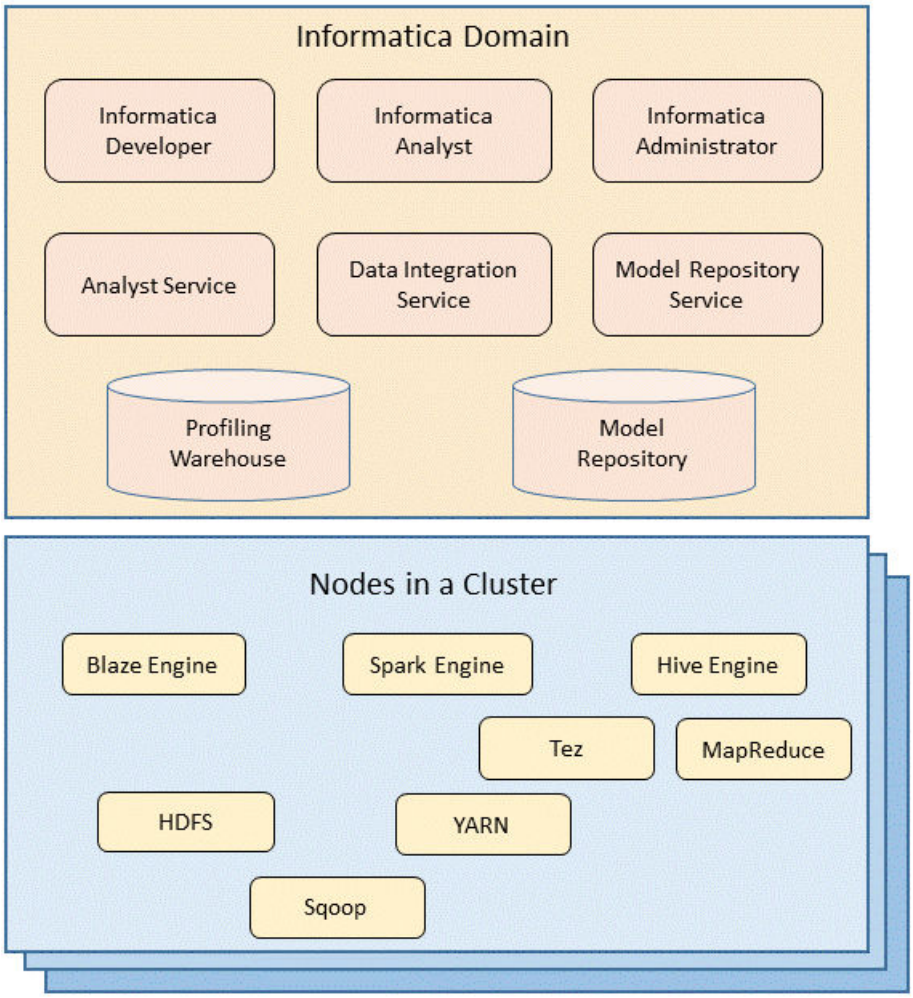
Introduction to Big Data Management Administration

This chapter includes the following topics:

Big Data Management Component Architecture

The Big Data Management components include client tools, application services, repositories, and third-party tools that Big Data Management uses for a big data project. The specific components involved depend on the task you perform.

The following image shows the components of Big Data Management:



Clients and Tools

Based on your product license, you can use multiple Informatica tools and clients to manage big data projects.

Use the following tools to manage big data projects:

Informatica Administrator

Monitor the status of profile, mapping, and MDM Big Data Relationship Management jobs on the Monitoring tab of the Administrator tool. The Monitoring tab of the Administrator tool is called the Monitoring tool. You can also design a Vibe Data Stream workflow in the Administrator tool.

Informatica Analyst

Create and run profiles on big data sources, and create mapping specifications to collaborate on projects and define business logic that populates a big data target with data.

Informatica Developer

Create and run profiles against big data sources, and run mappings and workflows on the Hadoop cluster from the Developer tool.

Application Services

Big Data Management uses application services in the Informatica domain to process data.

Big Data Management uses the following application services:

Analyst Service

The Analyst Service runs the Analyst tool in the Informatica domain. The Analyst Service manages the connections between service components and the users that have access to the Analyst tool.

Data Integration Service

The Data Integration Service can process mappings in the native environment or push the mapping for processing to the Hadoop cluster in the Hadoop environment. The Data Integration Service also retrieves metadata from the Model repository when you run a Developer tool mapping or workflow. The Analyst tool and Developer tool connect to the Data Integration Service to run profile jobs and store profile results in the profiling warehouse.

Model Repository Service

The Model Repository Service manages the Model repository. The Model Repository Service connects to the Model repository when you run a mapping, mapping specification, profile, or workflow.

Repositories

Big Data Management uses repositories and other databases to store data related to connections, source metadata, data domains, data profiling, data masking, and data lineage. Big Data Management uses application services in the Informatica domain to access data in repositories.

Big Data Management uses the following databases:

Model repository

The Model repository stores profiles, data domains, mapping, and workflows that you manage in the Developer tool. The Model repository also stores profiles, data domains, and mapping specifications that you manage in the Analyst tool.

Profiling warehouse

The Data Integration Service runs profiles and stores profile results in the profiling warehouse.

Hadoop Environment

Big Data Management can connect to clusters that run different Hadoop distributions. Hadoop is an open-source software framework that enables distributed processing of large data sets across clusters of machines. You might also need to use third-party software clients to set up and manage your Hadoop cluster.

Big Data Management can connect to Hadoop as a data source and push job processing to the Hadoop cluster. It can also connect to HDFS, which enables high performance access to files across the cluster. It can connect to Hive, which is a data warehouse that connects to HDFS and uses SQL-like queries to run

MapReduce jobs on Hadoop, or YARN, which can manage Hadoop clusters more efficiently. It can also connect to NoSQL databases such as HBase, which is a database comprising key-value pairs on Hadoop that performs operations in real-time.

The Data Integration Service pushes mapping and profiling jobs to the Blaze, Spark, or Hive engine in the Hadoop environment.

Hadoop Utilities

Big Data Management uses third-party Hadoop utilities such as Sqoop to process data efficiently.

Sqoop is a Hadoop command line program to process data between relational databases and HDFS through MapReduce programs. You can use Sqoop to import and export data. When you use Sqoop, you do not need to install the relational database client and software on any node in the Hadoop cluster.

To use Sqoop, you must configure Sqoop properties in a JDBC connection and run the mapping in the Hadoop environment. You can configure Sqoop connectivity for relational data objects, customized data objects, and logical data objects that are based on a JDBC-compliant database. For example, you can configure Sqoop connectivity for the following databases:

- Aurora
- Greenplum
- IBM DB2
- IBM DB2 for z/OS
- Microsoft SQL Server
- Netezza
- Oracle
- Teradata

The Model Repository Service uses JDBC to import metadata. The Data Integration Service runs the mapping in the Hadoop run-time environment and pushes the job processing to Sqoop. Sqoop then creates map-reduce jobs in the Hadoop cluster, which perform the import and export job in parallel.

Specialized Sqoop Connectors

When you run mappings through Sqoop, you can use the following specialized connectors:

OraOop

You can use OraOop with Sqoop to optimize performance when you read data from or write data to Oracle. OraOop is a specialized Sqoop plug-in for Oracle that uses native protocols to connect to the Oracle database.

You can configure OraOop when you run Sqoop mappings on the Spark and Hive engines.

Teradata Connector for Hadoop (TDCH) Specialized Connectors for Sqoop

You can use the following TDCH specialized connectors for Sqoop when you read data from or write data to Teradata:

- Cloudera Connector Powered by Teradata
- Hortonworks Connector for Teradata (powered by the Teradata Connector for Hadoop)

Cloudera Connector Powered by Teradata and Hortonworks Connector for Teradata are specialized Sqoop plug-ins that Cloudera and Hortonworks provide for Teradata. These TDCH Sqoop Connectors use native protocols to connect to the Teradata database.

You can configure the Cloudera Connector Powered by Teradata and Hortonworks Connector for Teradata when you run Sqoop mappings on the Blaze and Spark engines.

Note: For information about running native Teradata mappings with Sqoop, see the *Informatica PowerExchange for Teradata Parallel Transporter API User Guide*.

Big Data Management Engines

When you run a big data mapping, you can choose to run the mapping in the native environment or a Hadoop environment. If you run the mapping in a Hadoop environment, the mapping will run on the Blaze engine, the Spark engine, or the Hive engine.

When you validate a mapping, you can validate it against one or all of the engines. The Developer tool returns validation messages for each engine.

You can then choose to run the mapping in the native environment or in the Hadoop environment. When you run the mapping in the native environment, the Data Integration Service processes the mapping logic. When you run the mapping in the Hadoop environment, the Data Integration Service uses a proprietary rule-based methodology to determine the best engine to run the mapping. The rule-based methodology evaluates the mapping sources and the mapping logic to determine the engine. The Data Integration Service translates the mapping logic into code that the engine can process, and it transfers the code to the engine.

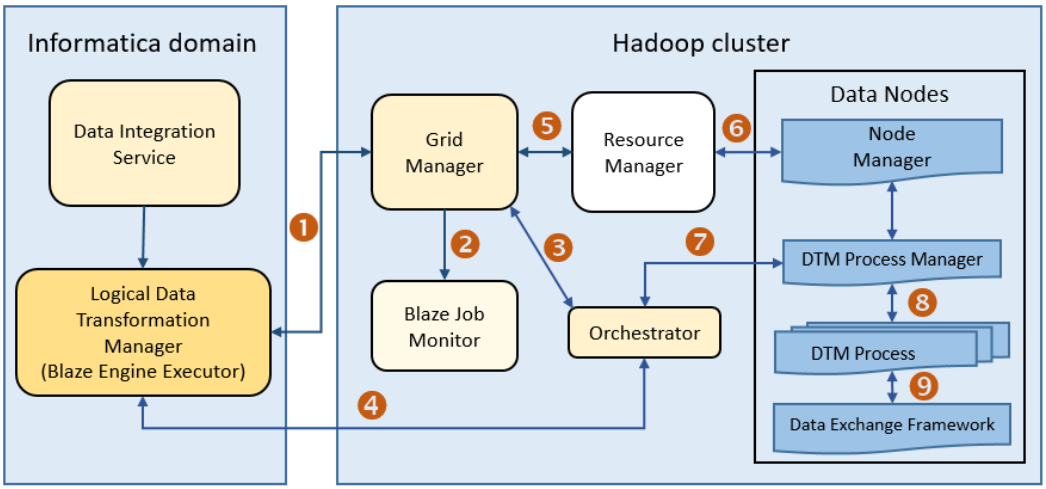
Blaze Engine Architecture

To run a mapping on the Informatica Blaze engine, the Data Integration Service submits jobs to the Blaze engine executor. The Blaze engine executor is a software component that enables communication between the Data Integration Service and the Blaze engine components on the Hadoop cluster.

The following Blaze engine components appear on the Hadoop cluster:

- Grid Manager. Manages tasks for batch processing.
- Orchestrator. Schedules and processes parallel data processing tasks on a cluster.
- Blaze Job Monitor. Monitors Blaze engine jobs on a cluster.
- DTM Process Manager. Manages the DTM Processes.
- DTM Processes. An operating system process started to run DTM instances.
- Data Exchange Framework. Shuffles data between different processes that process the data on cluster nodes.

The following image shows how a Hadoop cluster processes jobs sent from the Blaze engine executor:



The following events occur when the Data Integration Service submits jobs to the Blaze engine executor:

1. The Blaze Engine Executor communicates with the Grid Manager to initialize Blaze engine components on the Hadoop cluster, and it queries the Grid Manager for an available Orchestrator.
2. The Grid Manager starts the Blaze Job Monitor.
3. The Grid Manager starts the Orchestrator and sends Orchestrator information back to the LDTM.
4. The LDTM communicates with the Orchestrator.
5. The Grid Manager communicates with the Resource Manager for available resources for the Orchestrator.
6. The Resource Manager handles resource allocation on the data nodes through the Node Manager.
7. The Orchestrator sends the tasks to the DTM Processes through the DTM Process Manger.
8. The DTM Process Manager continually communicates with the DTM Processes.
9. The DTM Processes continually communicate with the Data Exchange Framework to send and receive data across processing units that run on the cluster nodes.

Application Timeline Server

The Hadoop Application Timeline Server collects basic information about completed application processes. The Timeline Server also provides information about completed and running YARN applications.

The Grid Manager starts the Application Timeline Server in the Yarn configuration by default.

The Blaze engine uses the Application Timeline Server to store the Blaze Job Monitor status. On Hadoop distributions where the Timeline Server is not enabled by default, the Grid Manager attempts to start the Application Timeline Server process on the current node.

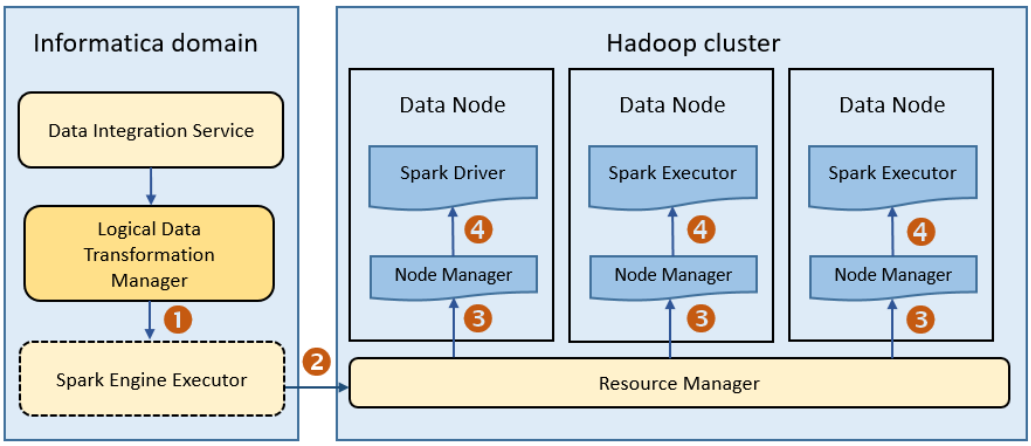
If you do not enable the Application Timeline Server on secured Kerberos clusters, the Grid Manager attempts to start the Application Timeline Server process in HTTP mode.

Spark Engine Architecture

The Data Integration Service can use the Spark engine on a Hadoop cluster to run Model repository mappings.

To run a mapping on the Spark engine, the Data Integration Service sends a mapping application to the Spark executor. The Spark executor submits the job to the Hadoop cluster to run.

The following image shows how a Hadoop cluster processes jobs sent from the Spark executor:



The following events occur when Data Integration Service runs a mapping on the Spark engine:

1. The Logical Data Transformation Manager translates the mapping into a Scala program, packages it as an application, and sends it to the Spark executor.
2. The Spark executor submits the application to the Resource Manager in the Hadoop cluster and requests resources to run the application.

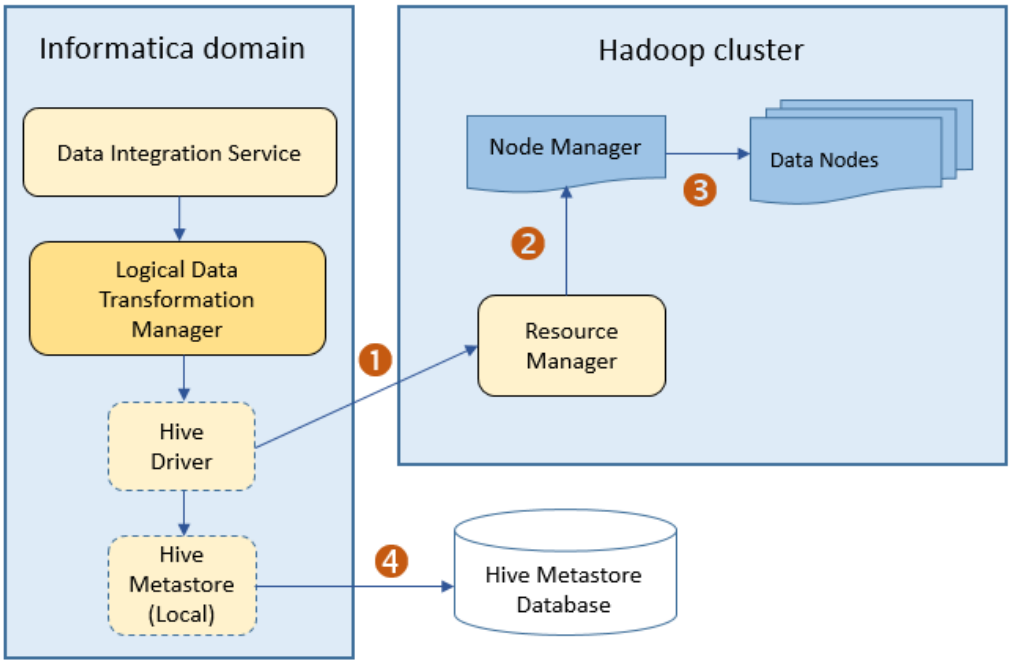
   Note: When you run mappings on the HDInsight cluster, the Spark executor launches a spark-submit script. The script requests resources to run the application.
3. The Resource Manager identifies the Node Managers that can provide resources, and it assigns jobs to the data nodes.
4. Driver and Executor processes are launched in data nodes where the Spark application runs.

Hive Engine Architecture

The Data Integration Service can use the Hive engine to run Model repository mappings or profiles on a Hadoop cluster.

To run a mapping or profile with the Hive engine, the Data Integration Service creates HiveQL queries based on the transformation or profiling logic. The Data Integration Service submits the HiveQL queries to the Hive driver. The Hive driver converts the HiveQL queries to MapReduce jobs, and then sends the jobs to the Hadoop cluster.

The following diagram shows the architecture of how a Hadoop cluster processes MapReduce jobs sent from the Hive driver:



The following events occur when the Hive driver sends jobs to the Hadoop cluster:

1. The Hive driver sends the MapReduce jobs to the Resource Manager in the Hadoop cluster.
2. The Resource Manager sends the jobs request to the Node Manager that retrieves a list of data nodes that can process the MapReduce jobs.
3. The Node Manager assigns MapReduce jobs to the data nodes.
4. The Hive driver also connects to the Hive metadata database through the Hive metastore to determine where to create temporary tables. The Hive driver uses temporary tables to process the data. The Hive driver removes temporary tables after completing the task.

CHAPTER 2

Authentication and Authorization

This chapter includes the following topics:

Authentication and Authorization Overview

You can configure security for Big Data Management and the Hadoop cluster to protect from threats inside and outside the network. Security for Big Data Management includes security for the Informatica domain and security for the Hadoop cluster.

Security for the Hadoop cluster includes the following areas:

Authentication

When the Informatica domain includes Big Data Management, user identities must be authenticated in the Informatica domain and the Hadoop cluster. Authentication for the Informatica domain is separate from authentication for the Hadoop cluster.

By default, Hadoop does not verify the identity of users. To authenticate user identities, you can configure the following authentication protocols on the cluster:

- Native authentication
- Lightweight Directory Access Protocol (LDAP)
- Kerberos, when the Hadoop distribution supports it
- Apache Knox Gateway

Big Data Management also supports Hadoop clusters that use a Microsoft Active Directory (AD) Key Distribution Center (KDC) or an MIT KDC.

Authorization

After a user is authenticated, a user must be authorized to perform actions. For example, a user must have the correct permissions to access the directories where specific data is stored to use that data in a mapping.

You can run mappings on a cluster that uses one of the following security management systems for authorization:

- HDFS permissions

- User impersonation
- Apache Ranger
- Apache Sentry
- HDFS Transparent Encryption

Data and metadata management

Data and metadata management involves managing data to track and audit data access, update metadata, and perform data lineage. Big Data Management supports Cloudera Navigator and Metadata Manager to manage metadata and perform data lineage.

Data security

Data security involves protecting sensitive data from unauthorized access. Big Data Management supports data masking with the Data Masking transformation in the Developer tool, Dynamic Data Masking, and Persistent Data Masking.

Operating system profiles

An operating system profile is a type of security that the Data Integration Service uses to run mappings. Use operating system profiles to increase security and to isolate the run-time environment for users. Big Data Management supports operating system profiles on all Hadoop distributions.

Support for Authentication Systems

Depending on the run-time engine that you use, you can run mappings on a Hadoop cluster that uses a supported security management system.

Hadoop clusters use a variety of security management systems for user authentication. The following table shows the run-time engines supported for the security management system installed on the Hadoop platform:

Hadoop Distribution	Apache Knox	Kerberos	LDAP	SASL
Amazon EMR	No support	No support	No support	No support
Azure HDInsight	No support	No support	No support	No support
Cloudera CDH	No support	- Native - Blaze - Spark - Hive	- Native - Blaze - Spark - Hive	- Native - Blaze - Spark - Hive
IBM BigInsights	- Native - Blaze - Spark - Hive	- Native - Blaze - Spark - Hive	No support	No support
Hortonworks HDP	- Native - Blaze - Spark - Hive	- Native - Blaze - Spark - Hive	- Native - Blaze - Spark - Hive	- Native - Blaze - Spark - Hive
MapR	No support	- Native - Blaze - Spark - Hive	No support	No support

Support for Authorization Systems

Depending on the run-time engine that you use, you can run mappings on a Hadoop cluster that uses a supported security management system.

Hadoop clusters use a variety of security management systems for user authorization. The following table shows the run-time engines supported for the security management system installed on the Hadoop platform:

Hadoop Distribution	Apache Ranger	Apache Sentry	HDFS Transparent Encryption	SSL/TLS	SQL Authorization
Amazon EMR	No support	No support	No support	No support	No support
Azure HDInsight	No support	No support	No support	No support	No support
Cloudera CDH	No support	- Native - Blaze - Spark - Hive	- Native - Blaze - Spark	- Native - Blaze - Spark - Hive	- Native - Blaze - Spark
IBM BigInsights	- Native - Blaze - Spark	No support	- Native - Blaze - Spark	No support	No support
Hortonworks HDP	- Native - Blaze - Spark **Note:** Also supports SQL authorization	No support	- Native - Blaze - Spark	- Native - Blaze - Spark - Hive	- Native - Blaze - Spark
MapR	No support	No support	No support	No support	No support

The combination of Apache Ranger and SQL authorization is supported on Hortonworks HDP only.

The combination of Apache Sentry and SQL authorization is supported on Cloudera 5.11, RedHat and SUSE, only.

Authentication

When the Informatica domain includes Big Data Management, user identities must be authenticated in the Informatica domain and the Hadoop cluster. Authentication for the Informatica domain is separate from authentication for the Hadoop cluster.

The authentication process verifies the identity of a user account.

By default, Hadoop does not authenticate users. Any user can be used in the Hadoop connection. Informatica recommends that you enable authentication for the cluster. If authentication is enabled for the cluster, the cluster authenticates the user account used for the Hadoop connection between Big Data Management and the cluster. For a higher level of security, you can set up Kerberos authentication for the cluster.

The Informatica domain uses one of the following authentication protocols:

Native authentication

The Informatica domain stores user credentials and privileges in the domain configuration repository and performs all user authentication within the Informatica domain.

Lightweight Directory Access Protocol (LDAP)

The LDAP directory service stores user accounts and credentials that are accessed over the network.

Kerberos authentication

Kerberos is a network authentication protocol that uses tickets to authenticate users and services in a network. Users are stored in the Kerberos principal database, and tickets are issued by a KDC.

User impersonation

User impersonation allows different users to run mappings on a Hadoop cluster that uses Kerberos authentication or connect to big data sources and targets that use Kerberos authentication.

Apache Knox Gateway

The Apache Knox Gateway is a REST API gateway that authenticates users and acts as a single access point for a Hadoop cluster.

For more information about how to enable authentication for the Hadoop cluster, see the documentation for your Hadoop distribution.

Authentication with Kerberos

Big Data Management and the Hadoop cluster can use Kerberos authentication to verify user accounts, when the Hadoop cluster supports Kerberos. You can use Kerberos authentication with the Informatica domain, with a supported Hadoop cluster, or with both.

The following distributions support Kerberos:

- Hortonworks HDP
- IBM Big Insights
- Cloudera CDH

Kerberos is a network authentication protocol that uses tickets to authenticate access to services and nodes in a network. Kerberos uses a Key Distribution Center (KDC) to validate the identities of users and services and to grant tickets to authenticated user and service accounts. Users and services are known as principals. The KDC has a database of principals and their associated secret keys that are used as proof of identity. Kerberos can use an LDAP directory service as a principal database.

You can integrate the Informatica domain with a Kerberos-enabled Hadoop cluster whether the domain is Kerberos-enabled or not.

The requirements for Kerberos authentication for the Informatica domain and for the Hadoop cluster:

Kerberos authentication for the Informatica domain

Kerberos authentication for the Informatica domain requires principals stored in a Microsoft Active Directory (AD) LDAP service. If the Informatica domain is Kerberos-enabled, you must use Microsoft AD for the KDC.

Kerberos authentication for the Hadoop cluster

Informatica supports Hadoop clusters that use an AD KDC or an MIT KDC.

When you enable Kerberos for Hadoop, each user and Hadoop service must be authenticated by the KDC. The cluster must authenticate the Data Integration Service User and, optionally, the Blaze user.

For more information about how to configure Kerberos for Hadoop, see the documentation for your Hadoop distribution.

The configuration steps required for Big Data Management to connect to a Hadoop cluster that uses Kerberos authentication depend on whether the Informatica domain uses Kerberos.

User Impersonation

User impersonation allows different users to run mappings in a Hadoop cluster that uses Kerberos authentication or connect to big data sources and targets that use Kerberos authentication.

The Data Integration Service uses its credentials to impersonate the user accounts designated in the Hadoop connection to connect to the Hadoop cluster or to start the Blaze engine.

When the Data Integration Service impersonates a user account to submit a mapping, the mapping can only access Hadoop resources that the impersonated user has permissions on. Without user impersonation, the Data Integration Service uses its credentials to submit a mapping to the Hadoop cluster. Restricted Hadoop resources might be accessible.

When the Data Integration service impersonates a user account to start the Blaze engine, the Blaze engine has the privileges and permissions of the user account used to start it.

Apache Knox Gateway

The Apache Knox Gateway is a REST API gateway that authenticates users and acts as a single access point for a Hadoop cluster.

Knox creates a perimeter around a Hadoop cluster. Without Knox, users and applications must connect directly to a resource in the cluster, which requires configuration on the client machines. A direct connection to resources exposes host names and ports to all users and applications and decreases the security of the cluster.

If the cluster uses Knox, applications use REST APIs and JDBC/ODBC over HTTP to connect to Knox. Knox authenticates the user and connects to a resource.

Authorization

Authorization controls what a user can do on a Hadoop cluster. For example, a user must be authorized to submit jobs to the Hadoop cluster.

You can use the following systems to manage authorization for Big Data Management:

HDFS permissions

By default, Hadoop uses HDFS permissions to determine what a user can do to a file or directory on HDFS. Additionally, Hadoop implements transparent data encryption in HDFS directories.

Apache Sentry

Sentry is a security plug-in that is used to enforce role-based authorization for data and metadata on a Hadoop cluster. Sentry can secure data and metadata at the table and column level. For example, Sentry can restrict access to columns that contain sensitive data and prevent unauthorized users from accessing the data.

Apache Ranger

Ranger is a security plug-in that is used to authenticate users of a Hadoop cluster. Ranger manages access to files, folders, databases, tables, and columns. When a user performs an action, Ranger verifies that the user meets the policy requirements and has the correct permissions on HDFS.

Fine-Grained SQL Authorization

SQL standards-based authorization enables database administrators to impose column-level authorization on Hive tables and views. A more fine-grained level of SQL standards-based authorization enables administrators to impose row and column level authorization. You can configure a Hive connection to observe fine-grained SQL standards-based authorization.

HDFS Permissions

HDFS permissions determine what a user can do to files and directories stored in HDFS. To access a file or directory, a user must have permission or belong to a group that has permission.

HDFS permissions are similar to permissions for UNIX or Linux systems. For example, a user requires the *r* permission to read a file and the *w* permission to write a file.

When a user or application attempts to perform an action, HDFS checks if the user has permission or belongs to a group with permission to perform that action on a specific file or directory.

Fine-Grained SQL Authorization for Hive

SQL standards-based authorization enables database administrators to impose fine-grained authorization on Hive tables and views when you read data from a Hive source.

When you choose to observe fine-grained SQL authentication in a Hive source, the mapping observes column-level restrictions on data access in the source.

Informatica supports fine-grained SQL authorization for Hive sources with all run-time engines. You can use the Ranger authorization plug-in when you enable fine-grained SQL authorization for mappings that run on a Hortonworks HDP cluster.

You can use the Sentry authorization plug-in when you enable fine-grained SQL authorization for mappings that run on a Cloudera cluster. When the mapping accesses Hive sources on a cluster that uses Sentry authorization and runs in native mode, you can use fine-grained SQL authorization on the column level if you configure hive.server2.proxy.user in the Hive JDBC connect string. In this case, the mapping uses the hive.server2.proxy.user value to access Hive sources. When you also configure the mappingImpersonationUserName property, then the mapping uses the mappingImpersonationUserName value to access Hive sources.

You can configure a mapping or a Hive connection to observe fine-grained SQL authorization.

Apache Ranger KMS and Cloudera Java KMS

Key Management Server or KMS is an open source key management service that supports HDFS data at rest encryption, or transparent encryption.

Ranger Key Management Store is a open source, scalable cryptographic key management service supporting HDFS data at rest encryption. For Cloudera CDH clusters, Cloudera provides a Key Management Server based on the Hadoop KeyProvider API to support Kerberos and TLS/SSL secure communication.

KMS enables the following functions:

Key management

You can create, update, or delete encryption key zones that control access to functionality.

Access control policies

You can administer access control policies for encryption keys. You can create or edit keys to control access by users to functionality.

Configuring Apache Ranger KMS or Cloudera Java KMS

Use the cluster administration utility to configure the KMS for access control.

1. Create a KMS user account for the Informatica user. Add the Informatica user to a new KMS repository, or to an existing KMS repository.

   The user corresponds to the Data Integration Service user or the Kerberos SPN user.
2. Grant permissions to the Informatica user.
3. Create and configure an encryption key.
4. Create an encryption zone that uses the encryption key you created.

   For example:

   ```
   hdfs dfs -mkdir /zone_encr_infa
   hdfs crypto -createZone -keyName infa_key -path /zone_encr_infa
   ```
5. Browse to the Custom KMS Site page and add the following properties:

   ```
   hadoop.kms.proxyuser.<user>.groups=*
   hadoop.kms.proxyuser.<user>.hosts=*
   hadoop.kms.proxyuser.<user>.users=*
   ```

   where <user> is the Informatica user name you configured in Step 1.
6. Update the following properties:

   ```
   hadoop.kms.proxyuser.<user>.hosts
   hadoop.kms.proxyuser.<user>.groups
   ```
7. Search for *proxyuser* in the KMS Configurations area. To register all Hadoop system users with the KMS, add the following properties:

   ```
   hadoop.kms.proxyuser.HTTP.hosts=*
   hadoop.kms.proxyuser.HTTP.users=*
   hadoop.kms.proxyuser.hive.hosts=*
   hadoop.kms.proxyuser.hive.users=*
   hadoop.kms.proxyuser.keyadmin.hosts=*
   hadoop.kms.proxyuser.keyadmin.users=*
   hadoop.kms.proxyuser.nn.hosts=*
   hadoop.kms.proxyuser.nn.users=*
   hadoop.kms.proxyuser.rm.hosts=*
   hadoop.kms.proxyuser.rm.users=*
   hadoop.kms.proxyuser.yarn.hosts=*
   hadoop.kms.proxyuser.yarn.users=*
   ```

Operating System Profiles

Use operating system profiles to increase security and to isolate the run-time environment for users. You can create and manage operating system profiles on the Security tab of the Administrator tool.

In the Hadoop run-time environment, the Data Integration Service pushes the processing to the Hadoop cluster and the Big Data Management engines run mappings with the operating system profile.

CHAPTER 3

Running Mappings on a Cluster with Kerberos Authentication

This chapter includes the following topics:

Running Mappings with Kerberos Authentication Overview

You can run mappings on a Hadoop cluster that uses MIT or Microsoft Active Directory (AD) Kerberos authentication. Kerberos is a network authentication protocol that uses tickets to authenticate access to services and nodes in a network.

The following distributions support Kerberos:

- Hortonworks HDP
- IBM Big Insights
- Cloudera CDH

If the Informatica domain uses Kerberos authentication, you must configure a one-way cross-realm trust to enable the Hadoop cluster to communicate with the Informatica domain. The Informatica domain uses Kerberos authentication on an AD service. The Hadoop cluster uses Kerberos authentication on an MIT service. Enable the cross-realm trust to enable the MIT service to communicate with the AD service.

Based on whether the Informatica domain uses Kerberos authentication or not, you might need to perform the following tasks to run mappings on a Hadoop cluster that uses Kerberos authentication:

- If you run mappings in a Hadoop environment, you must configure user impersonation to enable other users to run mappings on the Hadoop cluster.
- If you run mappings in the native environment, you must configure the mappings to read and process data from Hive sources that use Kerberos authentication.

- If you run a mapping that has Hive sources or targets, you must enable user authentication for the mapping on the Hadoop cluster.
- If you import metadata from Hive, complex file sources, and HBase sources, you must configure the Developer tool to use Kerberos credentials to access the Hive, complex file, and HBase metadata.

Running Mappings in a Kerberos-Enabled Hadoop Environment

To run mappings in a Kerberos-enabled Hadoop environment, you must configure the Kerberos configuration file, create user authentication artifacts, and configure Kerberos authentication properties for the Informatica domain.

The Kerberos configuration file `krb5.conf` contains configuration properties for the Kerberos realm. The one-way cross-realm trust enables the Informatica domain to communicate with the Hadoop cluster.

The Informatica domain uses Kerberos authentication on a Microsoft Active Directory service. The Hadoop cluster uses Kerberos authentication on an MIT Kerberos service. You set up a one-way cross-realm trust to enable the KDC for the MIT Kerberos service to communicate with the KDC for the Active Directory service. After you set up the cross-realm trust, you must configure the Informatica domain to enable mappings to run in the Hadoop cluster.

To run mappings on a cluster that uses Kerberos authentication, perform the following configuration tasks:

1. Set up the Kerberos configuration file.
2. When the Informatica domain uses Kerberos authentication, set up the one-way cross-realm trust.
3. Create matching operating system profile user names on each Hadoop cluster node.
4. Create the Service Principal Name and Keytab File in the Active Directory Server.
5. Specify the Kerberos authentication properties for the Data Integration Service.
6. Configure Execution Options for the Data Integration Service.

Step 1. Set Up the Kerberos Configuration File

Set the configuration properties for the Kerberos realm that the Hadoop cluster uses to krb5.conf on the machine on which the Data Integration Service runs.

If the Informatica domain does not use Kerberos authentication, set the properties for the MIT realm. If the Informatica domain uses Kerberos authentication, set the properties for the Active Directory realm and MIT realm.

`krb5.conf` is located in the `<Informatica Installation Directory>/java/jre/lib/security` directory.

1. Back up `krb5.conf` before you make any changes.
2. Edit `krb5.conf`.
3. In the *libdefaults* section, set the default_realm property required by Informatica.

   The default_realm is the name of the service realm for the Informatica domain. If the Informatica domain does not use Kerberos authentication, then the default realm must be the name of the AD service.

The following example shows the value if the Informatica domain does not use Kerberos authentication:

```
[libdefaults]
default_realm = HADOOP-AD-REALM
```

The following example shows the value if the Informatica domain uses Kerberos authentication:

```
[libdefaults]
default_realm = INFA-AD-REALM
```

4. In the *realms* section, set or add the properties required by Informatica.

   The following table lists the values to which you must set properties in the realms section:

Parameter	Value
kdc	Name of the host running a KDC server for that realm.
admin_server	Name of the Kerberos administration server.

   The following example shows the parameters for the Hadoop realm if the Informatica domain does not use Kerberos authentication:

```
[realms]
HADOOP-AD-REALM = {
        kdc = l23abcd134.hadoop-AD-realm.com
        admin_server = 123abcd124.hadoop-AD-realm.com
 }
```

   The following example shows the parameters for the Hadoop realm if the Informatica domain uses Kerberos authentication:

```
[realms]
INFA-AD-REALM = {
                kdc = abc123.infa-ad-realm.com
                admin_server = abc123.infa-ad-realm.com
                                                        }
HADOOP-MIT-REALM = {
                kdc = def456.hadoop-mit-realm.com
                admin_server = def456.hadoop-mit-realm.com
                                                        }
```

5. In the *domain_realms* section, map the domain name or host name to a Kerberos realm name. The domain name is prefixed by a period (.).

   The following example shows the parameters for the Hadoop domain_realm if the Informatica domain does not use Kerberos authentication:

```
[domain_realm]
 .hadoop_ad_realm.com = HADOOP-AD-REALM
  hadoop_ad_realm.com = HADOOP-AD-REALM
```

   The following example shows the parameters for the Hadoop domain_realm if the Informatica domain uses Kerberos authentication:

```
[domain_realm]
.infa_ad_realm.com = INFA-AD-REALM
  infa_ad_realm.com = INFA-AD-REALM
.hadoop_mit_realm.com = HADOOP-MIT-REALM
  hadoop_mit_realm.com = HADOOP-MIT-REALM
```

6. Copy the `krb5.conf` file to the following locations on the machine that hosts the Data Integration Service:

   - `<Informatica installation directory>/services/shared/security/`
   - `<Informatica installation directory>/java/jre/lib/security`

The following example shows the content of `krb5.conf` with the required properties for an Informatica domain that does not use Kerberos authentications:

```
[libdefaults]
default_realm = HADOOP-AD-REALM

[realms]
HADOOP-AD-REALM = {
        kdc = l23abcd134.hadoop-ad-realm.com
        admin_server = 123abcd124.hadoop-ad-realm.com
 }

[domain_realm]
 .hadoop_ad_realm.com = HADOOP-AD-REALM
  hadoop_ad_realm.com = HADOOP-AD-REALM
```

The following example shows the content of `krb5.conf` with the required properties for an Informatica domain that uses Kerberos authentication:

```
[libdefaults]
default_realm = INFA-AD-REALM

[realms]
INFA-AD-REALM = {
                kdc = abc123.infa-ad-realm.com
                admin_server = abc123.infa-ad-realm.com
                                                        }
HADOOP-MIT-REALM = {
                kdc = def456.hadoop-mit-realm.com
                admin_server = def456.hadoop-mit-realm.com
                                                        }

[domain_realm]
.infa_ad_realm.com = INFA-AD-REALM
  infa_ad_realm.com = INFA-AD-REALM
.hadoop_mit_realm.com = HADOOP-MIT-REALM
  hadoop_mit_realm.com = HADOOP-MIT-REALM
```

Step 2. Set up the Cross-Realm Trust

Perform this step when the Informatica domain uses Kerberos authentication.

Set up a one-way cross-realm trust to enable the KDC for the MIT Kerberos server to communicate with the KDC for the Active Directory server. When you set up the one-way cross-realm trust, the Hadoop cluster can authenticate the Active Directory principals.

To set up the cross-realm trust, you must complete the following steps:

1. Configure the Active Directory server to add the local MIT realm trust.
2. Configure the MIT server to add the cross-realm principal.
3. Translate principal names from the Active Directory realm to the MIT realm.

Configure the Microsoft Active Directory Server

Add the MIT KDC host name and local realm trust to the Active Directory server.

To configure the Active Directory server, complete the following steps:

1. Enter the following command to add the MIT KDC host name:

   ```
   ksetup /addkdc <mit_realm_name> <kdc_hostname>
   ```

   For example, enter the command to add the following values:

   ```
   ksetup /addkdc HADOOP-MIT-REALM def456.hadoop-mit-realm.com
   ```

2. Enter the following command to add the local realm trust to Active Directory:

```
netdom trust <mit_realm_name> /Domain:<ad_realm_name> /add /realm /
passwordt:<TrustPassword>
```

For example, enter the command to add the following values:

```
netdom trust HADOOP-MIT-REALM /Domain:INFA-AD-REALM /add /realm /passwordt:trust1234
```

3. Enter the following commands based on your Microsoft Windows environment to set the proper encryption type:
For Microsoft Windows 2008, enter the following command:

```
ksetup /SetEncTypeAttr <mit_realm_name> <enc_type>
```

For Microsoft Windows 2003, enter the following command:

```
ktpass /MITRealmName <mit_realm_name> /TrustEncryp <enc_type>
```

Note: The enc_type parameter specifies AES, DES, or RC4 encryption. To find the value for enc_type, see the documentation for your version of Windows Active Directory. The encryption type you specify must be supported on both versions of Windows that use Active Directory and the MIT server.

Configure the MIT Server

Configure the MIT server to add the cross-realm krbtgt principal. The krbtgt principal is the principal name that a Kerberos KDC uses for a Windows domain.

Enter the following command in the kadmin.local or kadmin shell to add the cross-realm krbtgt principal:

```
kadmin:  addprinc -e "<enc_type_list>" krbtgt/<mit_realm_name>@<MY-AD-REALM.COM>
```

The enc_type_list parameter specifies the types of encryption that this cross-realm krbtgt principal will support. The krbtgt principal can support either AES, DES, or RC4 encryption. You can specify multiple encryption types. However, at least one of the encryption types must correspond to the encryption type found in the tickets granted by the KDC in the remote realm.

For example, enter the following value:

```
kadmin:  addprinc -e "rc4-hmac:normal des3-hmac-sha1:normal" krbtgt/HADOOP-MIT-
REALM@INFA-AD-REALM
```

Translate Principal Names from the Active Directory Realm to the MIT Realm

To translate the principal names from the Active Directory realm into local names within the Hadoop cluster, you must configure the hadoop.security.auth_to_local property in the core-site.xml file on all the machines in the Hadoop cluster.

For example, set the following property in core-site.xml on all the machines in the Hadoop cluster:

```
<property>
  <name>hadoop.security.auth_to_local</name>
  <value>
    RULE:[1:$1@$0](^.*@INFA-AD-REALM$)s/^(.*)@INFA-AD-REALM$/$1/g
    RULE:[2:$1@$0](^.*@INFA-AD-REALM$)s/^(.*)@INFA-AD-REALM$/$1/g
    DEFAULT
  </value>
</property>
```

Step 3. Create Matching Operating System Profile Names

Create matching operating system profile user names on the machine that runs the Data Integration Service and each Hadoop cluster node to run Informatica mapping jobs.

For example, if user joe runs the Data Integration Service on a machine, you must create the user joe with the same operating system profile on each Hadoop cluster node.

Open a UNIX shell and enter the following UNIX command to create a user with the user name joe.

Step 4. Create the Principal Name and Keytab Files in the Active Directory Server

Create an SPN in the KDC database for Microsoft Active Directory service that matches the user name of the user that runs the Data Integration Service. Create a keytab file for the SPN on the machine on which the KDC server runs. Then, copy the keytab file to the machine on which the Data Integration Service runs.

You do not need to use the Informatica Kerberos SPN Format Generator to generate a list of SPNs and keytab file names. You can create your own SPN and keytab file name.

To create an SPN and Keytab file in the Active Directory server, complete the following steps:

Create a user in the Microsoft Active Directory Service.

Login to the machine on which the Microsoft Active Directory Service runs and create a user with the same name as the user you created in "Step 3. Create Matching Operating System Profile Names" on page 28.

Create an SPN associated with the user.

Use the following guidelines when you create the SPN and keytab files:

- The user principal name (UPN) must be the same as the SPN.
- Enable delegation in Microsoft Active Directory.
- Use the ktpass utility to create an SPN associated with the user and generate the keytab file.

  For example, enter the following command:

  ```
  ktpass -out infa_hadoop.keytab -mapuser joe -pass tempBG@2008 -princ joe/
  domain12345@INFA-AD-REALM -crypto all
  ```

  Note: The `-out` parameter specifies the name and path of the keytab file. The `-mapuser` parameter is the user to which the SPN is associated. The `-pass` parameter is the password for the SPN in the generated keytab. The `-princ` parameter is the SPN.

Step 5. Specify the Kerberos Authentication Properties for the Data Integration Service

In the Data Integration Service properties, configure the properties that enable the Data Integration Service to connect to a Hadoop cluster that uses Kerberos authentication. Use the Administrator tool to set the Data Integration Service properties.

Description	Property
Hadoop Kerberos Service Principal Name	Service Principal Name (SPN) of the Data Integration Service to connect to a Hadoop cluster that uses Kerberos authentication. Not required for the MapR distribution.
Hadoop Kerberos Keytab	The file path to the Kerberos keytab file on the machine on which the Data Integration Service runs. Not required for the MapR distribution.

Step 6. Configure the Execution Options for the Data Integration Service

To determine whether the Data Integration Service runs jobs in separate operating system processes or in one operating system process, configure the Launch Job Options property. Use the Administrator tool to configure the execution options for the Data Integration Service.

1. Click **Edit** to edit the **Launch Job Options** property in the execution options for the Data Integration Service properties.
2. Choose the launch job option.
   - If you configure the Data Integration Service to launch jobs as a separate process, you must specify the location of the `krb5.conf` file in the Java Virtual Manager (JVM) Options as a custom property in the Data Integration Service process. `krb5.conf` is located in the following directory:`<Informatica Installation Directory>/java/jre/lib/security`.
   - If you configure the Data Integration Service to launch jobs in the service process, you must specify the location of krb5.conf in the **Java Command Line Options** property in the Advanced Properties of the Data Integration Service process. Use the following syntax:

     ```
     -Djava.security.krb5.conf=<Informatica installation directory>/java/jre/lib/
     security/krb5.conf
     ```

User Impersonation with Kerberos Authentication

You can enable different users to run mappings in a Hadoop cluster that uses Kerberos authentication or connect to big data sources and targets that use Kerberos authentication. To enable different users to run mappings or connect to big data sources and targets, you must configure user impersonation.

You can configure user impersonation for the native or Hadoop environment.

Before you configure user impersonation, you must complete the following prerequisites:

- Complete the tasks for running mappings in a Kerberos-enabled Hadoop environment.
- Configure Kerberos authentication for the native or Hadoop environment.
- If the Hadoop cluster uses MapR, create a proxy directory for the user who will impersonate other users.

If the Hadoop cluster does not use Kerberos authentication, you can specify a user name in the Hadoop connection to enable the Data Integration Service to impersonate that user.

If the Hadoop cluster uses Kerberos authentication, you must specify a user name in the Hadoop connection.

User Impersonation in the Hadoop Environment

To enable different users to run mapping and workflow jobs on a Hadoop cluster that uses Kerberos authentication, you must configure user impersonation in the Hadoop environment.

For example, you want to enable user Bob to run mappings and workflows on the Hadoop cluster that uses Kerberos authentication.

To enable user impersonation, you must complete the following steps:

1. In the Active Directory, enable delegation for the Service Principal Name for the Data Integration Service to enable Bob to run Hadoop jobs.

2. If the service principal name (SPN) is different from the impersonation user, grant read permission on Hive tables to the SPN user.
3. Specify Bob as the user name in the Hadoop connection.

User Impersonation in the Native Environment

To enable different users to run mappings that read or processes data from big data sources or targets that use Kerberos authentication, configure user impersonation for the native environment.

To enable user impersonation, you must complete the following steps:

1. Specify Kerberos authentication properties for the Data Integration Service.
2. Configure the execution options for the Data Integration Service.

Step 1. Specify the Kerberos Authentication Properties for the Data Integration Service

In the Data Integration Service properties, configure the properties that enable the Data Integration Service to connect to a Hadoop cluster that uses Kerberos authentication. Use the Administrator tool to set the Data Integration Service properties.

Description	Property
Hadoop Kerberos Service Principal Name	Service Principal Name (SPN) of the Data Integration Service to connect to a Hadoop cluster that uses Kerberos authentication. Not required for the MapR distribution.
Hadoop Kerberos Keytab	The file path to the Kerberos keytab file on the machine on which the Data Integration Service runs. Not required for the MapR distribution.

Step 2. Configure the Execution Options for the Data Integration Service

To determine whether the Data Integration Service runs jobs in separate operating system processes or in one operating system process, configure the **Launch Job Options** property. Use the Administrator tool to configure the execution options for the Data Integration Service.

1. Click **Edit** to edit the **Launch Job Options** property in the execution options for the Data Integration Service properties.
2. Choose the launch job option.
   - If you configure the Data Integration Service to launch jobs as a separate process, you must specify the location of the krb5.conf in the Java Virtual Manager (JVM) Options as a custom property in the Data Integration Service process. krb5.conf is located in the following directory:`<Informatica Installation Directory>/java/jre/lib/security`.
   - If you configure the Data Integration Service to launch jobs in the service process, you must specify the location of krb5.conf in the **Java Command Line Options** property in the Advanced Properties of the Data Integration Service process. Use the following syntax:

     ```
     -Djava.security.krb5.conf=<Informatica installation directory>/java/jre/lib/
     security/krb5.conf
     ```

     .

Running Mappings in the Native Environment

To read and process data from Hive, HBase, or HDFS sources that use Kerberos authentication, you must configure Kerberos authentication for mappings in the native environment.

To read and process data from Hive, HBase, or HDFS sources, perform the following steps:

1. Complete the tasks for running mappings in a Kerberos-enabled Hadoop environment.
2. Complete the tasks for running mappings in the Hadoop environment when Informatica uses Kerberos authentication.
3. Create matching operating system profile user names on the machine that runs the Data Integration Service and each Hadoop cluster node used to run Informatica mapping jobs.
4. Create an Active Directory user that matches the operating system profile user you created in step 3.
5. Create an SPN associated with the user.
   Use the following guidelines when you create the SPN and keytab files:
   - The UPN must be the same as the SPN.
   - Enable delegation in Active Directory.
   - Use the ktpass utility to create an SPN associated with the user and generate the keytabs file. For example, enter the following command:

     ```
     ktpass -out infa_hadoop.keytab -mapuser joe -pass tempBG@2008 -princ joe/
     domain12345@HADOOP-AD-REALM -crypto all
     ```

     The `-out` parameter specifies the name and path of the keytab file. The `-mapuser` parameter is the user to which the SPN is associated. The `-pass` parameter is the password for the SPN in the generated keytab. The `-princ` parameter is the SPN.

Configure the Analyst Service

To use the Analyst Service with a Hadoop cluster that uses Kerberos authentication, configure the Analyst Service to use the Kerberos ticket that the Data Integration Service uses.

In the Administrator tool, select the Analyst Service. In the **Processes** tab, edit the Advanced Properties to add the following value to the JVM Command Line Options field: `DINFA_HADOOP_DIST_DIR=<Informatica installation directory>/services/shared/hadoop/<hadoop_distribution>`.

CHAPTER 4

Cluster Configuration

This chapter includes the following topics:

Cluster Configuration Overview

A cluster configuration is an object in the domain that contains configuration information about the Hadoop cluster. The cluster configuration enables the Data Integration Service to push mapping logic to the Hadoop environment.

Import configuration properties from the Hadoop cluster to create a cluster configuration. You can import directly from the cluster or from an archive file that the Hadoop administrator creates. When you perform the import, the cluster configuration wizard can create Hadoop, HBase, HDFS, and Hive connections to access the Hadoop environment. If you choose to create the connections, the wizard also associates the configuration object with the connections.

The cluster configuration displays properties in configuration sets that are based on *-site.xml files on the cluster. You can override the property values, and you can create user-defined properties based on your requirements. When property values change on the cluster, you can refresh the cluster configuration, either directly from the cluster or from an archive file.

The Developer tool requires the configuration properties to access the Hadoop cluster at design-time. You can generate a configuration archive file and extract it on the Developer tool machine.

Consider the following high-level process to manage cluster configurations:

1. Import the cluster configuration. Import properties associated with *-site.xml files from the cluster. Choose to create connections that require the cluster configuration.
2. Edit the cluster configuration. Override imported property values and add user-defined properties.
3. Make files available to the Developer tool. Generate a .zip archive from the cluster configuration and extract it on each Developer tool machine.

4. Refresh the cluster configuration. When property values change on the cluster, refresh the cluster configuration to import the changes. After you refresh the cluster configuration, generate updated files available for the Developer tool.

Cluster Configuration and Connections

When you create a cluster configuration, you can choose to create connections. All Hadoop connections used to run a mapping must be associated with a cluster configuration.

If you choose to create connections, the **Cluster Configuration** wizard associates the cluster configuration with each connection that it creates. The wizard creates the following connections:

- Hive
- HBase
- Hadoop
- HDFS

The wizard uses the following naming convention when it creates connections: <connection type>_<cluster configuration name>, such as Hive_ccMapR.

If you do not choose to create connections, you must manually create them and associate the cluster configuration with them.

Copying a Connection to Another Domain

When you copy a Hadoop, HDFS, HBase or Hive connection that is associated with a cluster configuration to another domain, you must first create a cluster configuration in the target domain.

Create a cluster configuration with the same name as the one that is associated with the connection in the source domain.

Note: When you create a cluster configuration in the domain, use the **Cluster Configuration** wizard. Informatica does not recommend importing the archive file into the domain to create a cluster configuration. If you import the archive file into the domain, the user-defined properties are converted to imported properties. When you subsequently refresh the cluster configuration, the refresh operation replaces the values of properties with cluster property values, and removes properties that do not exist on the cluster.

1. Identify a connection to copy, and note the name of the cluster configuration that is associated with it.
2. In the target domain, create a cluster configuration of the same name.
3. Choose not to create connections with creation of the cluster configuration.
4. Copy the connection to the target domain.

The connection has an associated cluster configuration.

Cluster Configuration Views

You can manage cluster configurations on the **Connections** tab of the Administrator tool.

When you highlight a cluster configuration on the **Connections** tab, you can view the cluster configuration details on the right pane. The cluster configuration displays the following components:

Active Properties view

Displays general properties and all run-time properties and values. You can view the following properties:

- Imported properties with imported values and any overridden values.
- User-defined properties and values. User-defined property values appear as overridden values.

Overridden Properties view

Displays only properties with overridden values, including imported properties and user-defined properties.

Permissions view

Configure permissions to perform actions on cluster configurations.

Actions menu

From the Actions menu, you can perform the following actions:

- Refresh the cluster configuration from the Hadoop cluster.
- Export the configuration to an archive file, required by the Developer tool to access cluster metadata at design-time.

You can create, edit, and delete properties from the **Active Properties** view and the **Overridden Properties** view.

Note: The **Active Properties** and **Overridden Properties** views display configuration sets with names based on the associated *-site.xml file on the cluster. For example, the properties from the cluster core-site.xml file appear under the configuration set name core_site_xml.

Active Properties View

The **Active Properties** view displays all cluster properties, both imported and user-defined.

The **Active Properties** view contains general properties and all run-time properties. General properties include the cluster configuration name, ID, description, distribution type, and the last date of refresh. Run-time properties are organized into configuration sets based on the corresponding *-site.xml files on the cluster. For example, the hive-site.xml configuration set contains all of the properties and values imported from the hive-site.xml file on the cluster.

The cluster configuration can contain the following types of run-time properties:

Imported properties

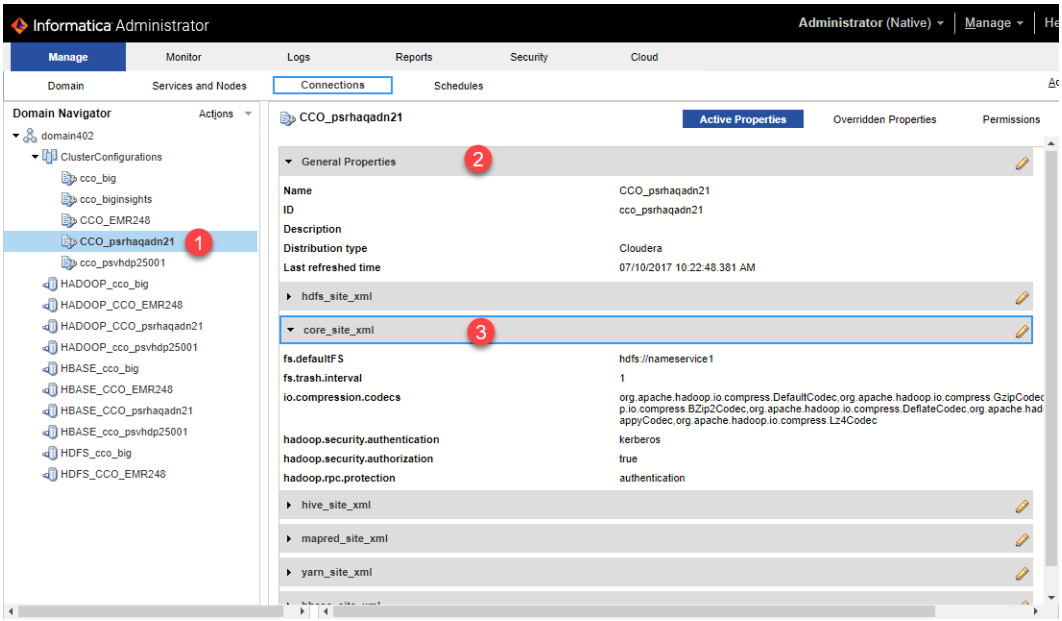
Properties and values imported from the cluster or file. You can override property values based on your requirements. Some cluster configuration properties contain sensitive information, such as passwords. The Service Manager masks the value of sensitive properties with asterisk characters when you import or refresh the cluster configuration. The masked values appear in the Administrator tool and in infacmd results.

User-defined properties

You can create user-defined properties based on processing requirements. When you create a user-defined property, the value appears as an overridden value.

Active properties are properties that the Data Integration Service uses at run time. Each expanded configuration set of the **Active Properties** view displays these active values. If a property has an overridden value, the Data Integration Service uses the overridden value as the active value. If the property does not have an overridden value, the Data Integration Service uses the imported value as the active value. To see the imported value of a property that is overridden, click the edit icon.

The following image shows cluster configurations in the **Domain Navigator**.



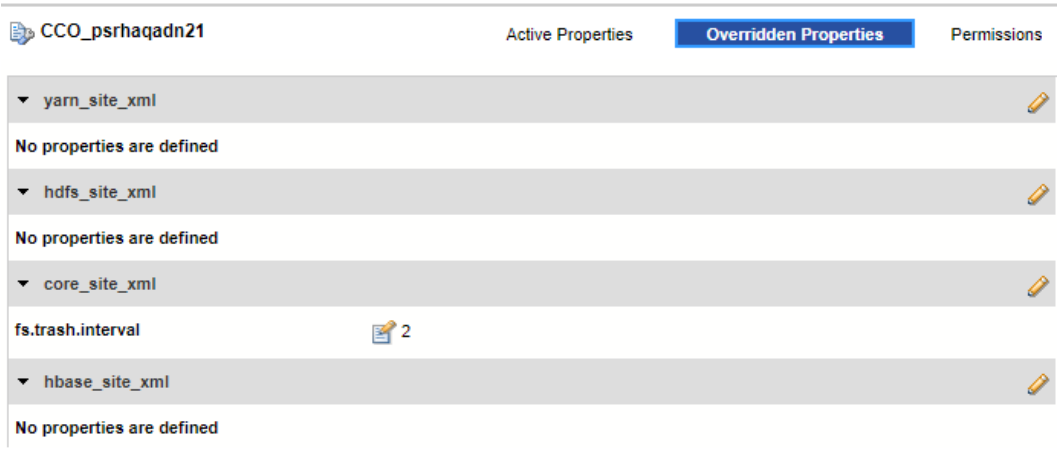
1. The **Cluster Configurations** node in the **Domain Navigator** displays the cluster configurations in the domain.
2. The right pane shows the general properties and configuration sets. The General Properties set is expanded to show general property values.
3. The core-site.xml configuration set is expanded to show the properties that it contains.

Overridden Properties View

The **Overridden Properties** view displays only properties with overridden values.

The **Overridden Properties** view includes user-defined properties and imported properties that you overrode. The values that appear in the view are the active values. To see imported values, click the edit icon.

The following image shows a property in the core-site.xml configuration set with an overridden value of 2:



Note that configuration sets that do not contain overrides display a message indicating that no properties are defined.

Create the Cluster Configuration

Import the cluster information into the domain. When you import cluster information, you import values from *-site.xml files to create a domain object called a cluster configuration.

Choose one of the following options to import cluster properties:

Import from cluster

When you import directly from the cluster, you enter cluster connection information. The Service Manager uses the information to connect to the cluster and get cluster configuration properties.

Note: You can import directly from Azure HDInsight, Cloudera CDH, Hortonworks HDP, and IBM BigInsights clusters.

Import from file

When you import from a file, you browse to an archive file that the Hadoop administrator created. Use this option if the Hadoop administrator requires you to do so.

Note: If you import from a MapR or Amazon EMR cluster, you must import from a file.

Before You Import

Before you can import the cluster configuration, you must get information from the Hadoop administrator, based on the method of import.

If you import directly from the cluster, contact the Hadoop administrator to get cluster connection information. If you import from a file, get an archive file of exported cluster information.

For more information about required cluster information, see the *Big Data Management Hadoop Integration Guide*.

Note: To import from Amazon EMR or MapR, you must import from an archive file.

Importing a Cluster Configuration from the Cluster

When you import the cluster configuration directly from the cluster, you provide information to connect to the cluster.

Get cluster connection information from the Hadoop administrator.

1. From the **Connections** tab, click the **ClusterConfigurations** node in the Domain Navigator.
2. From the Actions menu, select **New** > **Cluster Configuration**.

   The **Cluster Configuration** wizard opens.
3. Configure the following properties:

Property	Description
Cluster configuration name	Name of the cluster configuration.
Description	Optional description of the cluster configuration.
Distribution type	The cluster Hadoop distribution type.
Method to import the cluster configuration	Choose **Import from cluster**.
Create connections	Choose to create Hadoop, HDFS, Hive, and HBase connections. If you choose to create connections, the Cluster Configuration wizard associates the cluster configuration with each connection that it creates. If you do not choose to create connections, you must manually create them and associate the cluster configuration with them. **Important:** When the wizard creates the Hive connection, it populates the Metadata Connection String and the Data Access Connection String properties with the value from the hive.metastore.uris property. If the Hive metastore and HiveServer2 are running on different nodes, you must update the Metadata Connection String to point to the HiveServer2 host.

   The cluster properties appear.
4. Configure the following properties:

Property	Description
Host	IP address of the cluster manager.
Port	Port of the cluster manager.
User ID	Cluster user ID.
Password	Password for the user.
Cluster name	Name of the cluster. Use the display name if the cluster manager manages multiple clusters. If you do not provide a cluster name, the wizard imports information based on the default cluster.

5. Click **Next** and verify the cluster configuration information on the summary page.

Importing a Cluster Configuration from a File

You can import properties from an archive file to create a cluster configuration.

Before you import from the cluster, you must get the archive file from the Hadoop administrator.

1. From the **Connections** tab, click the **ClusterConfigurations** node in the Domain Navigator.
2. From the Actions menu, select **New** > **Cluster Configuration**.

   The **Cluster Configuration** wizard opens.
3. Configure the following properties:

Property	Description
Cluster configuration name	Name of the cluster configuration.
Description	Optional description of the cluster configuration.
Distribution type	The cluster Hadoop distribution type.
Method to import the cluster configuration	Choose **Import from file** to import properties from an archive file.
Create connections	Choose to create Hadoop, HDFS, Hive, and HBase connections. If you choose to create connections, the Cluster Configuration wizard associates the cluster configuration with each connection that it creates. If you do not choose to create connections, you must manually create them and associate the cluster configuration with them. **Important:** When the wizard creates the Hive connection, it populates the Metadata Connection String and the Data Access Connection String properties with the value from the hive.metastore.uris property. If the Hive metastore and HiveServer2 are running on different nodes, you must update the Metadata Connection String to point to the HiveServer2 host.

4. Click **Browse** to select a file. Select the file and click **Open**.
5. Click **Next** and verify the cluster configuration information on the summary page.

Edit the Cluster Configuration

You can add user-defined properties, override imported property values, and delete properties within a configuration set.

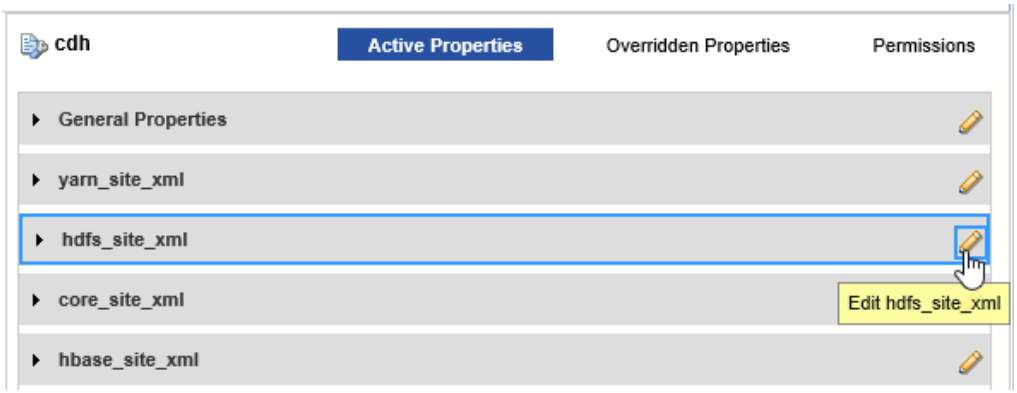
To edit the cluster configuration, you can access the **Edit** dialog box for a configuration set on the **Active Properties** view or the **Overridden Properties** view. Within the **Edit** dialog box, you can use the filter control to find properties.

Filtering Cluster Configuration Properties

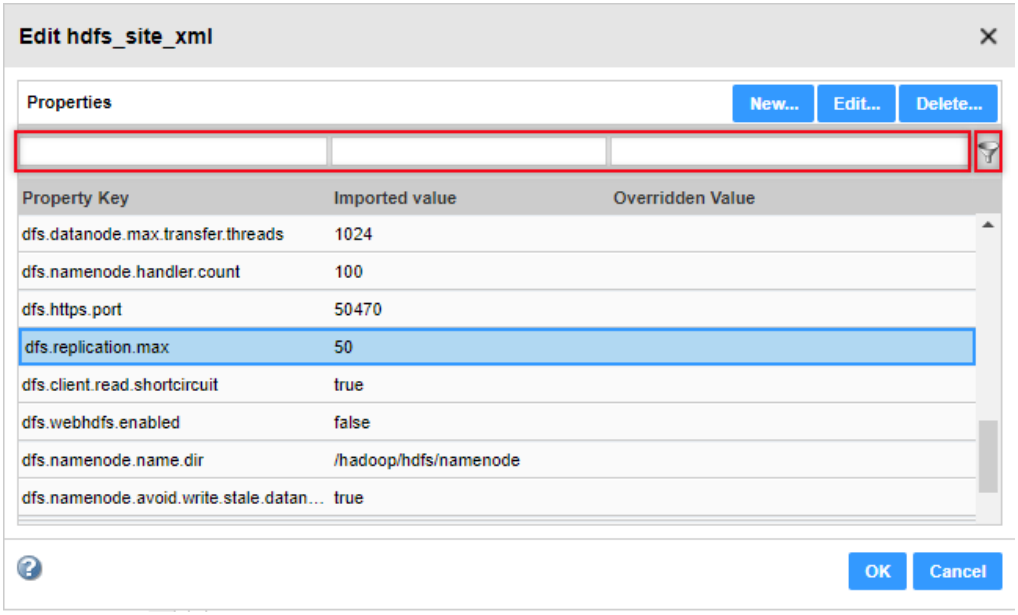
You can search for properties within a configuration set by using the filter controls.

You can filter properties in the **Active Properties** view or the **Overridden Properties** view. You might want to filter properties when a configuration set contains a large number of properties.

1. In the **Active Properties** view or the **Overridden Properties** view, expand a configuration set.
2. Click the **Edit** icon on the name bar of the configuration set that you want to edit.
   The following image shows the **Edit** icon for the hdfs-site.xml configuration set:



3. Enter text in the filter text entry pane above any column, and then click the filter icon. You can search by property, imported value, overridden value.
   The following image shows the filter text entry panes and the filter icon:



Overriding Imported Properties

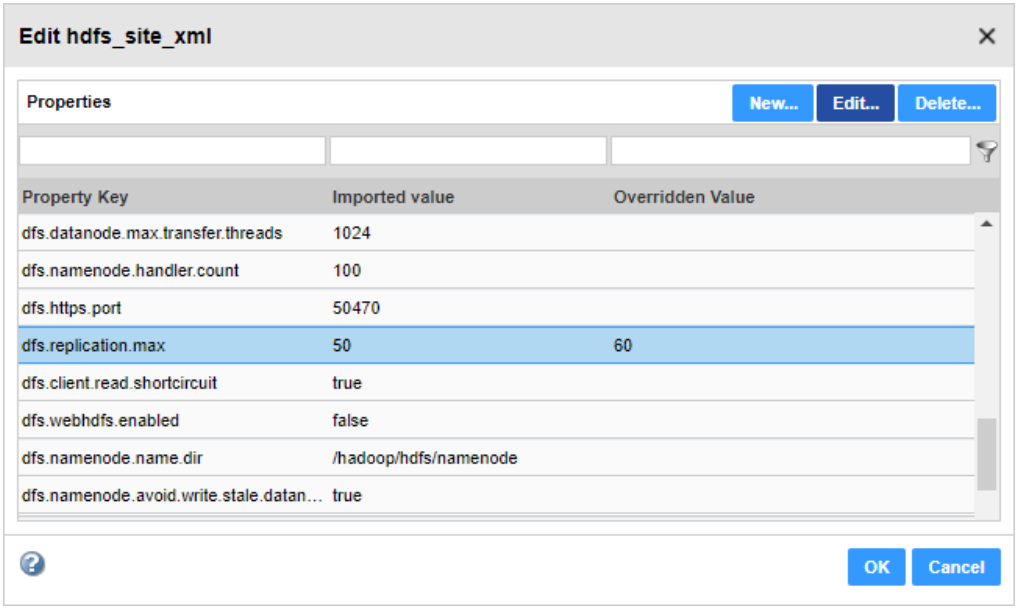
You can override property values or you can update overrides from the **Active Properties** view or the **Overridden Properties** view.

1. Expand the configuration set containing the property that you want to edit.
2. Click the **Edit** icon on the name bar of the configuration set that you want to edit.

   The **Edit** dialog box opens.
3. Optionally, use the filter controls to find a property.
4. Select the property to edit and click **Edit**.

   The **Edit Property** dialog box opens.
5. Enter a value in the **Overridden Value** pane, and click **OK**.

   The following image shows the overridden value in the **Edit** dialog box:

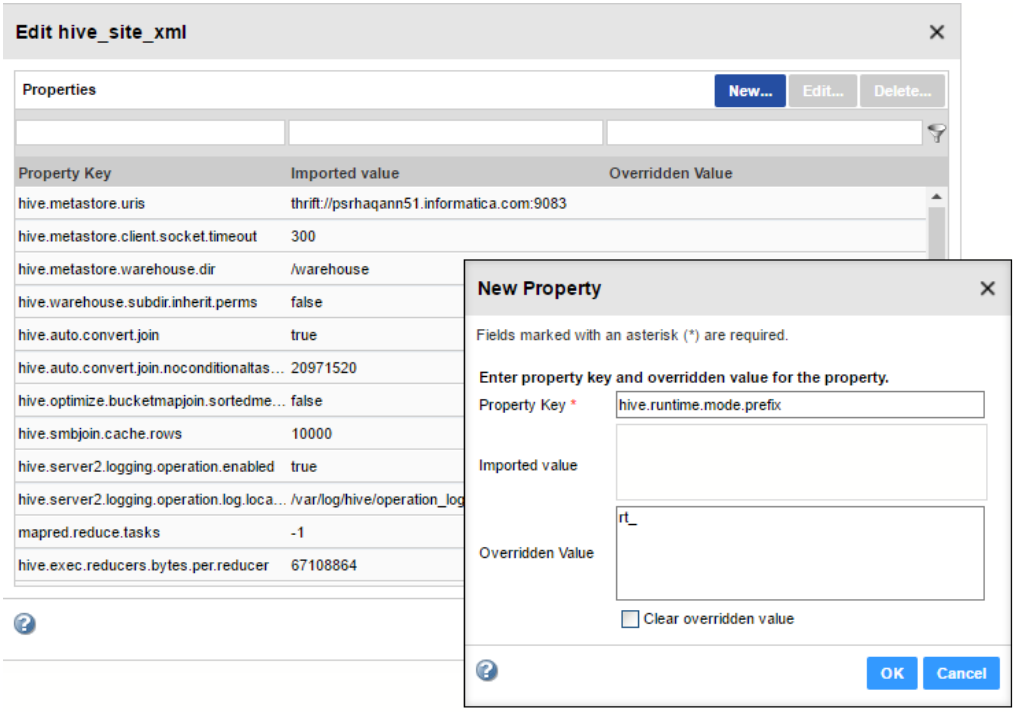


Creating User-Defined Properties

You can create user-defined properties in the **Active Properties** view or the **Overridden Properties** view. When you create a user-defined property, you configure an overridden value. You cannot configure an imported value in a user-defined property.

You can create a user-defined property based on your requirements.

1. Expand the configuration set where you want to create a property.
2. Click the **Edit** icon on the name bar of the configuration set that you want to edit.
   The **Edit** dialog box opens.
3. Click **New**.
   The **New Property** dialog box opens.



4. Configure the following properties:

Property	Description
Property Key	Name of the property that you want to enter.
Overridden Value	The property value. To clear the contents of this field, select **Clear overridden value**.

Important: If you create a property with the same name as a property that exists in a different configuration set, the Data Integration Service might use either property at run time, leading to unpredictable results.

5. Click **OK**.

Deleting Cluster Configuration Properties

You can delete imported and user-defined properties from a configuration set.

1. Select a cluster configuration to edit.
2. In the **Active Properties** view or the **Overridden Properties** view, expand a configuration set.
3. Click the **Edit** icon on the name bar of the configuration set that you want to edit.

   The configuration set expands to show its contents.
4. Optionally, use the filter control at the top of the **Property Key** column to filter the properties.
5. Select the property and click **Delete**.

Note: Imported properties that you delete will be restored if you refresh the cluster configuration.

Generate Configuration Files

The Developer tool requires configuration files to access cluster metadata at design-time. To provide access to the configuration files, you generate a cluster configuration archive file and extract it on the Developer tool machine.

To generate a configuration file, you export the contents of a cluster configuration to a .zip archive file. The archive contains .xml files based on the configuration sets in the cluster configuration.

The export process maintains the active values within the cluster configuration. For example, if you override an imported property value, the .xml file will contain the overridden value instead of the original imported value.

When you export the cluster configuration contents, select one of the following options:

Export with sensitive properties

When you export with sensitive properties, the Service Manager creates a .zip archive file that contains all the property-values pairs, including sensitive values, such as passwords. Values of sensitive properties are not masked. You can export the cluster configuration with sensitive properties if you have write permission on the cluster configuration.

Export without sensitive properties

When you export without sensitive properties, the Service Manager creates a .zip archive file that contains all the property-value pairs that do not have sensitive values. The sensitive properties are not required on the Developer tool machine. You can export the configuration files without sensitive properties if you have read permission on the cluster configuration.

Important: To maintain security, Informatica recommends that you generate archive files for the Developer tool *without* sensitive properties. Export *with* sensitive properties for administrative purposes only. An exception to note is that when you access Hive sources on S3 and use a non-EMR cluster for processing, generate the archive file with sensitive properties.

After you generate the configuration files, you must extract them on each Developer tool machine that accesses the Hadoop environment.

Exporting a Cluster Configuration

Export the cluster configuration from the Administrator tool to an archive file, and then extract the archive on the Developer tool machine.

1. Expand the **Cluster Configuration** node in the Domain Navigator.
2. Select the cluster configuration that you want to export.
3. From the Actions menu, select one of the following options:
   - **Export with sensitive properties**
   - **Export without sensitive properties**
4. The Administrator tool assigns a default name to the archive file using the name of the cluster configuration and a datetime string. For example, when the cluster configuration is named `CC1`: `CC1_2017-07-24-21-39-45.zip`

   Accept this name, or click **Save As**, name the file, and browse to a directory location to save the file and click **Save**.

   The Service Manager creates a .zip archive file that contains all properties and active values in the cluster configuration.

5. Copy the .zip file to the Developer tool machine and extract the contents to the following location:

   ```
   <Informatica installation directory>\clients\DeveloperClient\hadoop\<Hadoop distribution>
   \conf
   ```

Cluster Configuration Export Frequently Asked Questions

After I export a cluster configuration, can I use the archive file as the source to a cluster configuration on another domain?

When you create a cluster configuration in the domain, use the **Cluster Configuration** wizard. Informatica does not recommend importing the archive file into the domain to create a cluster configuration. If you import the archive file into the domain, the user-defined properties are converted to imported properties. When you subsequently refresh the cluster configuration, the refresh operation replaces the values of properties with cluster property values, and removes properties that do not exist on the cluster.

Refresh the Cluster Configuration

When property values change on the cluster, refresh the cluster configuration to import the changes. Similar to when you import the cluster configuration, you can refresh it either directly or from a .zip or .tar archive file.

You can refresh the cluster configuration from the Actions menu.

The refresh operation updates the cluster configuration in the following ways:

- Unedited values of imported properties are refreshed with the value from the cluster.
- The refresh operation refreshes the contents of the cluster configuration and properties in the connections that are associated with it, such as fs.defaultFS. If you refresh the cluster configuration from an archive file that does not contain one of the required *-site.xml files, the refresh cluster configuration drops the configuration set. If a missing *-site.xml file contained properties that get propagated to a connection, the refresh operation succeeds, but the connection refresh fails.
- If you override an imported property value, the refresh operation refreshes the imported value and maintains the overridden value. The Administrator tool displays the updated imported value and the active override value.
- If you override an imported property value, and the property is subsequently removed from the cluster, the refresh operation converts the property to a user-defined property.
- User-defined properties that do not exist in the cluster are not affected.
- If you configure a user-defined property that is subsequently configured on the cluster, the refresh operation converts the user-defined property to an imported property. The user-defined property value becomes an overridden value of the imported property.
- If you deleted an imported property, the refresh operation imports the property again if it exists on the cluster.

Note: After you refresh the cluster you must generate the configuration files for the Developer tool.

Example - Cluster Configuration Refresh

The following example shows the process for importing a property added during a refresh operation after you create it as a user-defined property.

1. You add the following user-defined property to the cluster configuration:

```
<property>
  <name>hive.runtime.mode.prefix</name>
  <value>rt_</value>
  <description>prefixes the runtime output table by this string</description>
</property>
```

2. The Hadoop administrator adds the property to the cluster, but with a different value, as follows:

```
<property>
  <name>hive.runtime.mode.prefix</name>
  <value>runtime_</value>
  <description>prefixes the runtime output table by this string</description>
</property>
```

3. You refresh the cluster configuration, and the refresh operation performs the following tasks.
   a. Converts the user-defined property to an imported property.
   b. Maintains the user-defined "rt_" value as an overridden value.
   c. Imports the cluster value "runtime_" as the imported value.

Delete a Cluster Configuration

You cannot delete a cluster configuration that has associated connections. You can associate the connections with a different cluster configuration before you delete the cluster configuration.

You can also use the infacmd cluster DeleteConfguration command to delete connections when you delete the cluster configuration.

You must have write permission to delete a cluster configuration, and Manage Connections privilege to delete connections.

To delete a cluster configuration, click the **Actions** menu and select **Delete**.

CHAPTER 5

Cluster Configuration Privileges and Permissions

This chapter includes the following topics:

Privileges and Roles

Cluster configuration privileges and roles determine the actions that users can perform using the Administrator tool and the infacmd command line program.

The following privileges and roles are required to perform certain actions on the cluster configuration:

Domain Administration privilege group

A user assigned the Administrator role for the domain can configure cluster configurations.

Manage Connections privilege

Users or groups assigned the Manage Connections privilege can create, refresh, and delete cluster configurations. Users can also set and clear configuration properties.

Administrator Privilege

To log in to the Administrator tool, you must have the Access Informatica Administrator domain privilege. If you have the Access Informatica Administrator privilege on a cluster configuration, but do not have the Manage Connections privilege that grants the ability to modify the cluster configuration, then you can view the cluster configuration properties. You can also export the cluster configuration without sensitive properties. You cannot edit or refresh the cluster configuration, set or clear configuration properties, export the cluster configuration with sensitive properties, or delete the cluster configuration.

Permissions

Permissions control the level of access that a user or group has for a cluster configuration.

You can configure permissions for a cluster configuration in the Administrator tool and using infacmd.

Any cluster configuration permission that is assigned to a user or group in one tool also applies in the other tool. For example, you grant GroupA permission on ConfigurationA using the Informatica command line interface. GroupA has permission on ConfigurationA in the Developer tool also.

The following Informatica components use the cluster configuration permissions:

- Administrator tool. Enforces read, write, execute, and grant permissions on cluster configurations.
- Informatica command line interface. Enforces read, write, execute, and grant permissions on cluster configurations.
- Developer tool. Enforces read, write, and execute permissions on cluster configurations.
- Data Integration Service. Enforces execute permissions when a user tries to preview data or run a mapping, scorecard, or profile.

Types of Cluster Configuration Permissions

You can assign different permission types to users to perform the following actions:

Permission Type	Action
Read	View the cluster configuration. Export the cluster configuration without sensitive properties.
Write	Edit and refresh the cluster configuration. Set and clear configuration properties. Export the cluster configuration with sensitive properties. Delete the cluster configuration. Users with write permission inherit read permission.
Execute	Run mappings in the Hadoop environment.
Grant	Grant permission on the cluster configuration to other users and groups. Users with grant permission inherit read permission.
All	Inherit read, write, execute, and grant permissions.
None	Remove permissions for the user.

Default Cluster Configuration Permissions

The domain administrator has all permissions on all cluster configurations. The user that creates a cluster configuration has read, write, execute, and grant permission for the cluster configuration. By default, all users have permission to view the cluster configuration name.

Assigning Permissions on a Cluster Configuration

When you assign permissions on a cluster configuration, you define the level of access a user or group has to the cluster configuration.

1. On the Manage tab, select the **Connections** view.
2. In the Navigator, select the cluster configuration.
3. In the contents panel, select the **Permissions** view.
4. Click the Groups or **Users** tab.
5. Click **Actions** > > **Assign Permission**.

The Assign Permissions dialog box displays all users or groups that do not have permission on the cluster configuration.

6. Enter the filter conditions to search for users and groups, and click the **Filter** button.
7. Select a user or group, and click **Next**.
8. Select **Allow** for each permission type that you want to assign.
9. Click **Finish**.

Viewing Permission Details on a Cluster Configuration

When you view permission details, you can view the origin of effective permissions.

1. On the Manage tab, select the **Connections** view.
2. In the Navigator, select the cluster configuration.
3. In the contents panel, select the **Permissions** view.
4. Click the **Groups** or **Users** tab.
5. Enter the filter conditions to search for users and groups, and click the **Filter** button.
6. Select a user or group and click **Actions** > **View Permission Details**.

   The View Permission Details dialog box appears. The dialog box displays direct permissions assigned to the user or group and direct permissions assigned to parent groups. In addition, permission details display whether the user or group is assigned the Administrator role which bypasses the permission check.
7. Click **Close**.
8. Or, click **Edit Permissions** to edit direct permissions.

Editing Permissions on a Cluster Configuration

You can edit direct permissions on a cluster configuration for a user or group. You cannot revoke inherited permissions or your own permissions.

1. **Note:** If you revoke direct permission on an object, the user or group might still inherit permission from a parent group or object.

   On the Manage tab, select the **Connections** view.
2. In the Navigator, select the cluster configuration.
3. In the contents panel, select the **Permissions** view.
4. Click the **Groups** or **Users** tab.
5. Enter the filter conditions to search for users and groups, and click the **Filter** button.
6. Select a user or group and click **Actions** > **Edit Direct Permissions**.

   The Edit Direct Permissions dialog box appears.
7. Choose to allow or revoke permissions.
   - Select **Allow** to assign a permission.
   - Clear **Allow** to revoke a single permission.
   - Select **Revoke** to revoke all permissions.

   You can view whether the permission is directly assigned or inherited by clicking **View Permission Details**.

8. Click **OK**.

CHAPTER 6

Queuing

This chapter includes the following topic:

Persisted Queues

The Data Integration Service uses persisted queues to store deployed mapping jobs and workflow mapping tasks. Persisted queuing protects against data loss if a node shuts down unexpectedly.

When you deploy a mapping job or workflow mapping task, the Data Integration Service moves these jobs directly to the persisted queue for that node. The job state is "Queued" in the Administrator tool contents panel. When resources are available, the Data Integration Service starts running the job.

Every node in a grid has one queue. Therefore, if the Data Integration Service shuts down unexpectedly, the queue does not fail over when the Data Integration Service fails over. The queue remains on the Data Integration Service machine, and the Data Integration Service resumes processing the queue when you restart it.

Note: While persisted queues help prevent data loss, you can still lose data if a node shuts down unexpectedly. In this case, all jobs in the "Running" state are marked as "Unknown." You must manually run these jobs again when the node restarts.

By default, each queue can hold 10,000 jobs at a time. When the queue is full, the Data Integration Service rejects job requests and marks them as failed. When the Data Integration Service starts running jobs in the queue, you can deploy additional jobs.

Persisted queuing is available for certain types of jobs, but not all. When you run a job that cannot be queued, the Data Integration Service immediately starts running the job. If there are not enough resources available, the job fails.

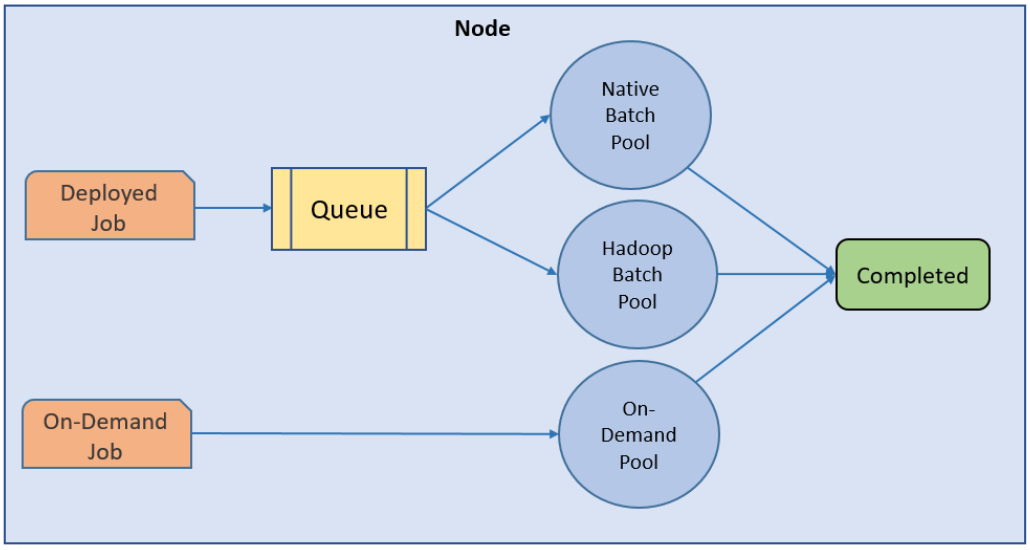
The following job types cannot be queued:

- Data previews
- Profiling jobs
- REST queries
- SQL queries
- Web service requests

Queuing Process

The Data Integration Service queues deployed jobs before running them in the native or Hadoop batch pool. On-demand jobs run immediately in the on-demand pool.

The following diagram shows the overall queuing and execution process:



When you deploy a mapping job or workflow mapping task, the Data Integration Service moves the job directly to the persisted queue for that node. If the queue is full, the Data Integration Service marks the job as failed.

You can cancel a job in the queue. A job is aborted if the node shuts down unexpectedly and the Data Integration Service is configured to discard all jobs in the queue upon restart.

When resources are available, the Data Integration Service moves the job to the execution pool and starts running the job. A deployed job runs in one of the following execution pools:

Native Batch Pool

Runs deployed native jobs.

Hadoop Batch Pool

Runs deployed Hadoop jobs.

You can cancel a running job, or the job may be aborted if the node shuts down unexpectedly. A job can also fail while running.

The Data Integration Service marks successful jobs as completed.

The Data Integration Service immediately starts running on-demand jobs. If you run more jobs than the **On-Demand Pool** can run concurrently, the extra jobs fail. You must manually run the jobs again when space is available.

The following table describes the mapping job states in the Administrator tool contents panel:

Job Status	Rules and Guidelines
Queued	The job is in the queue.
Running	The Data Integration Service is running the job.

Job Status	Rules and Guidelines
Completed	The job ran successfully.
Aborted	The job was flushed from the queue at restart or the node shut down unexpectedly while the job was running.
Failed	The job failed while running or the queue is full.
Canceled	The job was deleted from the queue or cancelled while running.
Unknown	The job status is unknown.

APPENDIX A

Connections

This appendix includes the following topics:

Connections

Define a Hadoop connection to run a mapping in the Hadoop environment. Depending on the sources and targets, define connections to access data in HBase, HDFS, Hive, or relational databases. You can create the connections using the Developer tool, Administrator tool, and infacmd.

You can create the following types of connections:

Hadoop connection

Create a Hadoop connection to run mappings in the Hadoop environment. If you select the mapping validation environment or the execution environment as Hadoop, select the Hadoop connection. Before you run mappings in the Hadoop environment, review the information in this guide about rules and guidelines for mappings that you can run in the Hadoop environment.

HBase connection

Create an HBase connection to access HBase. The HBase connection is a NoSQL connection.

HDFS connection

Create an HDFS connection to read data from or write data to the HDFS file system on a Hadoop cluster.

Hive connection

Create a Hive connection to access Hive as a source or target. You can access Hive as a source if the mapping is enabled for the native or Hadoop environment. You can access Hive as a target if the mapping runs on the Blaze or Hive engine.

JDBC connection

Create a JDBC connection and configure Sqoop properties in the connection to import and export relational data through Sqoop.

Note: For information about creating connections to other sources or targets such as social media web sites or Teradata, see the respective PowerExchange adapter user guide for information.

Hadoop Connection Properties

Use the Hadoop connection to configure mappings to run on a Hadoop cluster. A Hadoop connection is a cluster type connection. You can create and manage a Hadoop connection in the Administrator tool or the Developer tool. You can use infacmd to create a Hadoop connection. Hadoop connection properties are case sensitive unless otherwise noted.

Hadoop Cluster Properties

The following table describes the general connection properties for the Hadoop connection:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. Enter a string that you can use to identify the connection. The description cannot exceed 4,000 characters.
Cluster Configuration	The name of the cluster configuration associated with the Hadoop environment.

Common Properties

The following table describes the common connection properties that you configure for the Hadoop connection:

Property	Description
Impersonation User Name	Required if the Hadoop cluster uses Kerberos authentication. Hadoop impersonation user. The user name that the Data Integration Service impersonates to run mappings in the Hadoop environment. The Data Integration Service runs mappings based on the user that is configured. Refer the following order to determine which user the Data Integration Services uses to run mappings: 1. Operating system profile user. The mapping runs with the operating system profile user if the profile user is configured. If there is no operating system profile user, the mapping runs with the Hadoop impersonation user. 2. Hadoop impersonation user. The mapping runs with the Hadoop impersonation user if the operating system profile user is not configured. If the Hadoop impersonation user is not configured, the Data Integration Service runs mappings with the Data Integration Service user. 3. Informatica services user. The mapping runs with the operating user that starts the Informatica daemon if the operating system profile user and the Hadoop impersonation user are not configured.
Temporary Table Compression Codec	Hadoop compression library for a compression codec class name.
Codec Class Name	Codec class name that enables data compression and improves performance on temporary staging tables.
Hive Staging Database Name	Namespace for Hive staging tables. Use the name `default` for tables that do not have a specified database name. If you do not configure a namespace, the Data Integration Service uses the Hive database name in the Hive target connection to create staging tables.
Hadoop Engine Custom Properties	Custom properties that are unique to the Hadoop connection. You can specify multiple properties. Use the following format: `<property1>=<value>` To specify multiple properties use `&:` as the property separator. If more than one Hadoop connection is associated with the same cluster configuration, you can override configuration set property values. Use Informatica custom properties only at the request of Informatica Global Customer Support.

Reject Directory Properties

The following table describes the connection properties that you configure to the Hadoop Reject Directory.

Property	Description
Write Reject Files to Hadoop	If you use the Blaze engine to run mappings, select the check box to specify a location to move reject files. If checked, the Data Integration Service moves the reject files to the HDFS location listed in the property, Reject File Directory. By default, the Data Integration Service stores the reject files based on the RejectDir system parameter.
Reject File Directory	The directory for Hadoop mapping files on HDFS when you run mappings.

Hive Pushdown Configuration

The following table describes the connection properties that you configure to push mapping logic to the Hadoop cluster:

Property	Description
Environment SQL	SQL commands to set the Hadoop environment. The Data Integration Service executes the environment SQL at the beginning of each Hive script generated in a Hive execution plan. The following rules and guidelines apply to the usage of environment SQL: - Use the environment SQL to specify Hive queries. - Use the environment SQL to set the classpath for Hive user-defined functions and then use environment SQL or PreSQL to specify the Hive user-defined functions. You cannot use PreSQL in the data object properties to specify the classpath. The path must be the fully qualified path to the JAR files used for user-defined functions. Set the parameter hive.aux.jars.path with all the entries in infapdo.aux.jars.path and the path to the JAR files for user-defined functions. - You can use environment SQL to define Hadoop or Hive parameters that you want to use in the PreSQL commands or in custom queries. - If you use multiple values for the environment SQL, ensure that there is no space between the values.
Hive Warehouse Directory	Optional. The absolute HDFS file path of the default database for the warehouse that is local to the cluster. If you do not configure the Hive warehouse directory, the Hive engine first tries to write to the directory specified in the cluster configuration property `hive.metastore.warehouse.dir`. If the cluster configuration does not have the property, the Hive engine writes to the default directory `/user/hive/warehouse`.

Hive Configuration

The following table describes the connection properties that you configure for the Hive engine:

Property	Description
Engine Type	The engine that the Hadoop environment uses to run a mapping on the Hadoop cluster. You can choose MRv2 or Tez. You can select Tez if it is configured for the Hadoop cluster. Default is MRv2.

Blaze Configuration

The following table describes the connection properties that you configure for the Blaze engine:

Property	Description
Blaze Staging Directory	The HDFS file path of the directory that the Blaze engine uses to store temporary files. Verify that the directory exists. The YARN user, Blaze engine user, and mapping impersonation user must have write permission on this directory. Default is `/blaze/workdir`. If you clear this property, the staging files are written to the Hadoop staging directory `/tmp/blaze_<user name>`.
Blaze User Name	The owner of the Blaze service and Blaze service logs. When the Hadoop cluster uses Kerberos authentication, the default user is the Data Integration Service SPN user. When the Hadoop cluster does not use Kerberos authentication and the Blaze user is not configured, the default user is the Data Integration Service user.
Minimum Port	The minimum value for the port number range for the Blaze engine. Default is 12300.
Maximum Port	The maximum value for the port number range for the Blaze engine. Default is 12600.
YARN Queue Name	The YARN scheduler queue name used by the Blaze engine that specifies available resources on a cluster.
Blaze Job Monitor Address	The host name and port number for the Blaze Job Monitor. Use the following format: `<hostname>:<port>` Where - <hostname> is the host name or IP address of the Blaze Job Monitor server. - <port> is the port on which the Blaze Job Monitor listens for remote procedure calls (RPC). For example, enter: `myhostname:9080`
Blaze Service Custom Properties	Custom properties that are unique to the Blaze engine. To enter multiple properties, separate each name-value pair with the following text: `&:`. Use Informatica custom properties only at the request of Informatica Global Customer Support.

Spark Configuration

The following table describes the connection properties that you configure for the Spark engine:

Property	Description
Spark Staging Directory	The HDFS file path of the directory that the Spark engine uses to store temporary files for running jobs. The YARN user, Data Integration Service user, and mapping impersonation user must have write permission on this directory. By default, the temporary files are written to the Hadoop staging directory `/tmp/spark_<user name>`.
Spark Event Log Directory	Optional. The HDFS file path of the directory that the Spark engine uses to log events.

Property	Description
YARN Queue Name	The YARN scheduler queue name used by the Spark engine that specifies available resources on a cluster. The name is case sensitive.
Spark Execution Parameters	An optional list of configuration parameters to apply to the Spark engine. You can change the default Spark configuration properties values, such as `spark.executor.memory` or `spark.driver.cores`. Use the following format: `<property1>=<value>` To enter multiple properties, separate each name-value pair with the following text: `&:`

HDFS Connection Properties

Use a Hadoop File System (HDFS) connection to access data in the Hadoop cluster. The HDFS connection is a file system type connection. You can create and manage an HDFS connection in the Administrator tool, Analyst tool, or the Developer tool. HDFS connection properties are case sensitive unless otherwise noted.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes HDFS connection properties:

Property	Description
Name	Name of the connection. The name is not case sensitive and must be unique within the domain. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: `~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /`
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 765 characters.
Location	The domain where you want to create the connection. Not valid for the Analyst tool.
Type	The connection type. Default is Hadoop File System.
User Name	User name to access HDFS.
NameNode URI	The URI to access the storage system. You can find the value for `fs.defaultFS` in the `core-site.xml` configuration set of the cluster configuration.
Cluster Configuration	The name of the cluster configuration associated with the Hadoop environment.

Accessing Multiple Storage Types

Use the NameNode URI property in the connection parameters to connect to various storage types. The following table lists the storage type and the NameNode URI format for the storage type:

Storage	NameNode URI Format
HDFS	`hdfs://<namenode>:<port>` where: - `<namenode>` is the host name or IP address of the NameNode. - `<port>` is the port that the NameNode listens for remote procedure calls (RPC). `hdfs://<nameservice>` in case of NameNode high availability.
MapR-FS	`maprfs:///`

WASB in HDInsight	`wasb://<container_name>@<account_name>.blob.core.windows.net/<path>`
	where:
	- `<container_name>` identifies a specific Azure Storage Blob container.
	Note: `<container_name>` is optional.
	- `<account_name>` identifies the Azure Storage Blob object.
	Example:
	`wasb://infabdmoffering1storage.blob.core.windows.net/infabdmoffering1cluster/mr-history`

ADLS in HDInsight	`adl://home`

When you create a cluster configuration from an Azure HDInsight cluster, the cluster configuration uses either ADLS or WASB as the primary storage. You can edit the NameNode URI property in the HDFS connection to connect to a local HDFS location.

HBase Connection Properties

Use an HBase connection to access HBase. The HBase connection is a NoSQL connection. You can create and manage an HBase connection in the Administrator tool or the Developer tool. HBase connection properties are case sensitive unless otherwise noted.

The following table describes HBase connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select HBase.
Database Type	Type of database that you want to connect to. Select **HBase** to create a connection for an HBase table.
Cluster Configuration	The name of the cluster configuration associated with the Hadoop environment.

HBase Connection Properties for MapR-DB

Use an HBase connection to connect to a MapR-DB table. The HBase connection is a NoSQL connection. You can create and manage an HBase connection in the Administrator tool or the Developer tool. HBase connection properties are case sensitive unless otherwise noted.

The following table describes the HBase connection properties for MapR-DB:

Property	Description
Name	Name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Description of the connection. The description cannot exceed 4,000 characters.
Location	Domain where you want to create the connection.
Type	Connection type. Select **HBase**.
Database Type	Type of database that you want to connect to. Select **MapR-DB** to create a connection for a MapR-DB table.
Cluster Configuration	The name of the cluster configuration associated with the Hadoop environment.
MapR-DB Database Path	Database path that contains the MapR-DB table that you want to connect to. Enter a valid MapR cluster path. When you create an HBase data object for MapR-DB, you can browse only tables that exist in the MapR-DB path that you specify in the **Database Path** field. You cannot access tables that are available in sub-directories in the specified path. For example, if you specify the path as `/user/customers/`, you can access the tables in the `customers` directory. However, if the `customers` directory contains a sub-directory named `regions`, you cannot access the tables in the following directory: `/user/customers/regions`

Hive Connection Properties

Use the Hive connection to access Hive data. A Hive connection is a database type connection. You can create and manage a Hive connection in the Administrator tool, Analyst tool, or the Developer tool. Hive connection properties are case sensitive unless otherwise noted.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes Hive connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 4000 characters.
Location	The domain where you want to create the connection. Not valid for the Analyst tool.
Type	The connection type. Select Hive.
Connection Modes	Hive connection mode. Select **Access Hive as a source or target** to use Hive as a source or a target. **Note:** The **Use Hive to run mappings on a Hadoop cluster** mode is deprecated. To use the Hive driver to run mappings in the Hadoop cluster, use a Hadoop connection.
User Name	User name of the user that the Data Integration Service impersonates to run mappings on a Hadoop cluster. The user name depends on the JDBC connection string that you specify in the Metadata Connection String or Data Access Connection String for the native environment. If the Hadoop cluster runs Hortonworks HDP, you must provide a user name. If you use Tez to run mappings, you must provide the user account for the Data Integration Service. If you do not use Tez to run mappings, you can use an impersonation user account. If the Hadoop cluster uses Kerberos authentication, the principal name for the JDBC connection string and the user name must be the same. Otherwise, the user name depends on the behavior of the JDBC driver. With Hive JDBC driver, you can specify a user name in many ways and the user name can become a part of the JDBC URL. If the Hadoop cluster does not use Kerberos authentication, the user name depends on the behavior of the JDBC driver. If you do not specify a user name, the Hadoop cluster authenticates jobs based on the following criteria: - The Hadoop cluster does not use Kerberos authentication. It authenticates jobs based on the operating system profile user name of the machine that runs the Data Integration Service. - The Hadoop cluster uses Kerberos authentication. It authenticates jobs based on the SPN of the Data Integration Service. User Name will be ignored.
Password	Password for the user name.

Property	Description
Environment SQL	SQL commands to set the Hadoop environment. In native environment type, the Data Integration Service executes the environment SQL each time it creates a connection to a Hive metastore. If you use the Hive connection to run profiles on a Hadoop cluster, the Data Integration Service executes the environment SQL at the beginning of each Hive session. The following rules and guidelines apply to the usage of environment SQL in both connection modes: - Use the environment SQL to specify Hive queries. - Use the environment SQL to set the classpath for Hive user-defined functions and then use environment SQL or PreSQL to specify the Hive user-defined functions. You cannot use PreSQL in the data object properties to specify the classpath. The path must be the fully qualified path to the JAR files used for user-defined functions. Set the parameter hive.aux.jars.path with all the entries in infapdo.aux.jars.path and the path to the JAR files for user-defined functions. - You can use environment SQL to define Hadoop or Hive parameters that you want to use in the PreSQL commands or in custom queries. - If you use multiple values for the Environment SQL property, ensure that there is no space between the values. If you use the Hive connection to run profiles on a Hadoop cluster, the Data Integration service executes only the environment SQL of the Hive connection. If the Hive sources and targets are on different clusters, the Data Integration Service does not execute the different environment SQL commands for the connections of the Hive source or target.
SQL Identifier Character	The type of character used to identify special characters and reserved SQL keywords, such as WHERE. The Data Integration Service places the selected character around special characters and reserved SQL keywords. The Data Integration Service also uses this character for the **Support mixed-case identifiers** property.
Cluster Configuration	The name of the cluster configuration associated with the Hadoop environment.

Properties to Access Hive as Source or Target

The following table describes the connection properties that you configure to access Hive as a source or target:

Property	Description
JDBC Driver Class Name	Name of the Hive JDBC driver class. By default, the Apache Hive JDBC driver shipped with the distribution is considered. You can override the Apache Hive JDBC driver with a third-party Hive JDBC driver by specifying the driver class name.
Metadata Connection String	The JDBC connection URI used to access the metadata from the Hadoop server. You can use PowerExchange for Hive to communicate with a HiveServer service or HiveServer2 service. To connect to HiveServer, specify the connection string in the following format: `jdbc:hive2://<hostname>:<port>/<db>` Where - <hostname> is name or IP address of the machine on which HiveServer2 runs. - <port> is the port number on which HiveServer2 listens. - <db> is the database name to which you want to connect. If you do not provide the database name, the Data Integration Service uses the default database details. To connect to HiveServer 2, use the connection string format that Apache Hive implements for that specific Hadoop Distribution. For more information about Apache Hive connection string formats, see the Apache Hive documentation. For user impersonation, you must add `hive.server2.proxy.user=<xyz>` to the JDBC connection URI. If you do not configure user impersonation, the current user's credentials are used connect to the HiveServer2. If the Hadoop cluster uses SSL or TLS authentication, you must add `ssl=true` to the JDBC connection URI. For example: `jdbc:hive2://<hostname>:<port>/<db>;ssl=true` If you use self-signed certificate for SSL or TLS authentication, ensure that the certificate file is available on the client machine and the Data Integration Service machine. For more information, see the *Informatica Big Data Management Hadoop Integration Guide*.
Bypass Hive JDBC Server	JDBC driver mode. Select the check box to use the embedded JDBC driver mode. To use the JDBC embedded mode, perform the following tasks: - Verify that Hive client and Informatica services are installed on the same machine. - Configure the Hive connection properties to run mappings on a Hadoop cluster. If you choose the non-embedded mode, you must configure the Data Access Connection String. Informatica recommends that you use the JDBC embedded mode.

Property	Description
Observe Fine Grained SQL Authorization	When you select the option to observe fine-grained SQL authentication in a Hive source, the mapping observes row and column-level restrictions on data access. If you do not select the option, the Blaze run-time engine ignores the restrictions, and results include restricted data. Applicable to Hadoop clusters where Sentry or Ranger security modes are enabled.
Data Access Connection String	The connection string to access data from the Hadoop data store. To connect to HiveServer, specify the non-embedded JDBC mode connection string in the following format: `jdbc:hive2://<hostname>:<port>/<db>` Where - <hostname> is name or IP address of the machine on which HiveServer2 runs. - <port> is the port number on which HiveServer2 listens. - <db> is the database to which you want to connect. If you do not provide the database name, the Data Integration Service uses the default database details. To connect to HiveServer 2, use the connection string format that Apache Hive implements for the specific Hadoop Distribution. For more information about Apache Hive connection string formats, see the Apache Hive documentation. For user impersonation, you must add `hive.server2.proxy.user=<xyz>` to the JDBC connection URI. If you do not configure user impersonation, the current user's credentials are used connect to the HiveServer2. If the Hadoop cluster uses SSL or TLS authentication, you must add `ssl=true` to the JDBC connection URI. For example: `jdbc:hive2://<hostname>:<port>/<db>;ssl=true` If you use self-signed certificate for SSL or TLS authentication, ensure that the certificate file is available on the client machine and the Data Integration Service machine. For more information, see the *Informatica Big Data Management Hadoop Integration Guide*.

Properties to Run Mappings on a Hadoop Cluster

The following table describes the Hive connection properties that you configure when you want to use the Hive connection to run Informatica mappings on a Hadoop cluster:

Property	Description
Database Name	Namespace for tables. Use the name `default` for tables that do not have a specified database name.

Advanced Hive/Hadoop Properties	Configures or overrides Hive or Hadoop cluster properties in the `hive-site.xml` configuration set on the machine on which the Data Integration Service runs. You can specify multiple properties.
	Select **Edit** to specify the name and value for the property. The property appears in the following format:
	`<property1>=<value>`
	When you specify multiple properties, `&:` appears as the property separator.
	The maximum length for the format is 1 MB.
	If you enter a required property for a Hive connection, it overrides the property that you configure in the Advanced Hive/Hadoop Properties.
	The Data Integration Service adds or sets these properties for each map-reduce job. You can verify these properties in the JobConf of each mapper and reducer job. Access the JobConf of each job from the Jobtracker URL under each map-reduce job.
	The Data Integration Service writes messages for these properties to the Data Integration Service logs. The Data Integration Service must have the log tracing level set to log each row or have the log tracing level set to verbose initialization tracing.
	For example, specify the following properties to control and limit the number of reducers to run a mapping job:
	`mapred.reduce.tasks=2&:hive.exec.reducers.max=10`

Temporary Table Compression Codec	Hadoop compression library for a compression codec class name. You can choose None, Zlib, Gzip, Snappy, Bz2, LZO, or Custom. Default is None.
Codec Class Name	Codec class name that enables data compression and improves performance on temporary staging tables.

JDBC Connection Properties

You can use a JDBC connection to access tables in a database. You can create and manage a JDBC connection in the Administrator tool, the Developer tool, or the Analyst tool.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes JDBC connection properties:

Property	Description
Database Type	The database type.
Name	Name of the connection. The name is not case sensitive and must be unique within the domain. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: `~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /`
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 765 characters.
User Name	The database user name. If you configure Sqoop, Sqoop uses the user name that you configure in this field. If you configure the --username argument in a JDBC connection or mapping, Sqoop ignores the argument.
Password	The password for the database user name. If you configure Sqoop, Sqoop uses the password that you configure in this field. If you configure the --password argument in a JDBC connection or mapping, Sqoop ignores the argument.

JDBC Driver Class Name	Name of the JDBC driver class.
	The following list provides the driver class name that you can enter for the applicable database type:
	- DataDirect JDBC driver class name for Oracle:
	`com.informatica.jdbc.oracle.OracleDriver`
	- DataDirect JDBC driver class name for IBM DB2:
	`com.informatica.jdbc.db2.DB2Driver`
	- DataDirect JDBC driver class name for Microsoft SQL Server:
	`com.informatica.jdbc.sqlserver.SQLServerDriver`
	- DataDirect JDBC driver class name for Sybase ASE:
	`com.informatica.jdbc.sybase.SybaseDriver`
	- DataDirect JDBC driver class name for Informix:
	`com.informatica.jdbc.informix.InformixDriver`
	- DataDirect JDBC driver class name for MySQL:
	`com.informatica.jdbc.mysql.MySQLDriver`
	For more information about which driver class to use with specific databases, see the vendor documentation.

Property	Description

Connection String	Connection string to connect to the database. Use the following connection string:
	`jdbc:<subprotocol>:<subname>`
	The following list provides sample connection strings that you can enter for the applicable database type:
	- Connection string for DataDirect Oracle JDBC driver:
	`jdbc:informatica:oracle://<host>:<port>;SID=<value>`
	- Connection string for Oracle JDBC driver:
	`jdbc:oracle:thin:@//<host>:<port>:<SID>`
	- Connection string for DataDirect IBM DB2 JDBC driver:
	`jdbc:informatica:db2://<host>:<port>;DatabaseName=<value>`
	- Connection string for IBM DB2 JDBC driver:
	`jdbc:db2://<host>:<port>/<database_name>`
	- Connection string for DataDirect Microsoft SQL Server JDBC driver:
	`jdbc:informatica:sqlserver://<host>;DatabaseName=<value>`
	- Connection string for Microsoft SQL Server JDBC driver:
	`jdbc:sqlserver://<host>;DatabaseName=<value>`
	- Connection string for Netezza JDBC driver:
	`jdbc:netezza://<host>:<port>/<database_name>`
	- Connection string for Pivotal Greenplum driver:
	`jdbc:pivotal:greenplum://<host>:<port>;/database_name=<value>`
	- Connection string for Postgres Greenplum driver:
	`jdbc:postgressql://<host>:<port>/<database_name>`
	- Connection string for Teradata JDBC driver:
	`jdbc:teradata://<host>/database_name=<value>,tmode=<value>,charset=<value>`
	For more information about the connection string to use with specific drivers, see the vendor documentation.

Environment SQL	Optional. Enter SQL commands to set the database environment when you connect to the database. The Data Integration Service executes the connection environment SQL each time it connects to the database. **Note:** If you enable Sqoop, Sqoop ignores this property.
Transaction SQL	Optional. Enter SQL commands to set the database environment when you connect to the database. The Data Integration Service executes the transaction environment SQL at the beginning of each transaction. **Note:** If you enable Sqoop, Sqoop ignores this property.
SQL Identifier Character	Type of character that the database uses to enclose delimited identifiers in SQL queries. The available characters depend on the database type. Select (None) if the database uses regular identifiers. When the Data Integration Service generates SQL queries, the service does not place delimited characters around any identifiers. Select a character if the database uses delimited identifiers. When the Data Integration Service generates SQL queries, the service encloses delimited identifiers within this character. **Note:** If you enable Sqoop, Sqoop ignores this property.
Support Mixed-case Identifiers	Enable if the database uses case-sensitive identifiers. When enabled, the Data Integration Service encloses all identifiers within the character selected for the **SQL Identifier Character** property. When the **SQL Identifier Character** property is set to none, the **Support Mixed-case Identifiers** property is disabled. **Note:** If you enable Sqoop, Sqoop honors this property when you generate and execute a DDL script to create or replace a target at run time. In all other scenarios, Sqoop ignores this property.

Property	Description
Use Sqoop Connector	Enables Sqoop connectivity for the data object that uses the JDBC connection. The Data Integration Service runs the mapping in the Hadoop run-time environment through Sqoop. You can configure Sqoop connectivity for relational data objects, customized data objects, and logical data objects that are based on a JDBC-compliant database. Select **Sqoop v1.x** to enable Sqoop connectivity. Default is **None**.

Sqoop Arguments	Enter the arguments that Sqoop must use to connect to the database. Separate multiple arguments with a space.
	If you want to use Teradata Connector for Hadoop (TDCH) specialized connectors for Sqoop and run the mapping on the Blaze engine, define the TDCH connection factory class in the Sqoop arguments. The connection factory class varies based on the TDCH Sqoop Connector that you want to use.
	- To use the Cloudera Connector Powered by Teradata, configure the following Sqoop argument:
	`–Dsqoop.connection.factories=com.cloudera.connector.teradata.TeradataManagerFactory`
	- To use the Hortonworks Connector for Teradata (powered by the Teradata Connector for Hadoop), configure the following Sqoop argument:
	`–Dsqoop.connection.factories=org.apache.sqoop.teradata.TeradataManagerFactory`
	Note: You do not need to define the TDCH connection factory class in the Sqoop arguments if you run the mapping on the Spark engine.
	If you do not enter Sqoop arguments, the Data Integration Service constructs the Sqoop command based on the JDBC connection properties.
	On the Hive engine, to run a column profile on a relational data object that uses Sqoop, set the Sqoop argument m to 1. Use the following syntax:
	`-m 1`

Sqoop Connection-Level Arguments

In the JDBC connection, you can define the arguments that Sqoop must use to connect to the database. The Data Integration Service merges the arguments that you specify with the default command that it constructs based on the JDBC connection properties. The arguments that you specify take precedence over the JDBC connection properties.

If you want to use the same driver to import metadata and run the mapping, and do not want to specify any additional Sqoop arguments, select **Sqoop v1.x** from the **Use Sqoop Version** list and leave the **Sqoop Arguments** field empty in the JDBC connection. The Data Integration Service constructs the Sqoop command based on the JDBC connection properties that you specify.

However, if you want to use a different driver for run-time tasks or specify additional run-time Sqoop arguments, select **Sqoop v1.x** from the **Use Sqoop Version** list and specify the arguments in the **Sqoop Arguments** field.

You can configure the following Sqoop arguments in the JDBC connection:

driver

Defines the JDBC driver class that Sqoop must use to connect to the database.

Use the following syntax:

```
--driver <JDBC driver class>
```

For example, use the following syntax depending on the database type that you want to connect to:

- Aurora: `--driver com.mysql.jdbc.Driver`
- Greenplum: `--driver org.postgresql.Driver`
- IBM DB2: `--driver com.ibm.db2.jcc.DB2Driver`
- IBM DB2 z/OS: `--driver com.ibm.db2.jcc.DB2Driver`
- Microsoft SQL Server: `--driver com.microsoft.sqlserver.jdbc.SQLServerDriver`
- Netezza: `--driver org.netezza.Driver`
- Oracle: `--driver oracle.jdbc.driver.OracleDriver`
- Teradata: `--driver com.teradata.jdbc.TeraDriver`

connect

Defines the JDBC connection string that Sqoop must use to connect to the database. The JDBC connection string must be based on the driver that you define in the driver argument.

Use the following syntax:

```
--connect <JDBC connection string>
```

For example, use the following syntax depending on the database type that you want to connect to:

- Aurora: `--connect "jdbc:mysql://<host_name>:<port>/<schema_name>"`
- Greenplum: `--connect jdbc:postgresql://<host_name>:<port>/<database_name>`
- IBM DB2: `--connect jdbc:db2://<host_name>:<port>/<database_name>`
- IBM DB2 z/OS: `--connect jdbc:db2://<host_name>:<port>/<database_name>`
- Microsoft SQL Server: `--connect jdbc:sqlserver://<host_name>:<port or named_instance>;databaseName=<database_name>`
- Netezza: `--connect "jdbc:netezza://<database_server_name>:<port>/<database_name>;schema=<schema_name>"`
- Oracle: `--connect jdbc:oracle:thin:@<database_host_name>:<database_port>:<database_SID>`
- Teradata: `--connect jdbc:teradata://<host_name>/database=<database_name>`

direct

When you read data from or write data to Oracle, you can configure the direct argument to enable Sqoop to use OraOop. OraOop is a specialized Sqoop plug-in for Oracle that uses native protocols to connect to the Oracle database. When you configure OraOop, the performance improves.

You can configure OraOop when you run Sqoop mappings on the Spark and Hive engines.

Use the following syntax:

```
--direct
```

When you use OraOop, you must use the following syntax to specify multiple arguments:

```
-D<argument=value> -D<argument=value>
```

Note: If you specify multiple arguments and include a space character between -D and the argument name-value pair, Sqoop considers only the first argument and ignores the remaining arguments.

To direct a MapReduce job to a specific YARN queue, configure the following argument:

```
-Dmapred.job.queue.name=<YARN queue name>
```

If you do not direct the job to a specific queue, the Spark engine uses the default queue.

-Dsqoop.connection.factories

If you want to use Teradata Connector for Hadoop (TDCH) specialized connectors for Sqoop and run the mapping on the Blaze engine, you can configure the -Dsqoop.connection.factories argument. Use the argument to define the TDCH connection factory class that Sqoop must use. The connection factory class varies based on the TDCH Sqoop Connector that you want to use.

- To use the Cloudera Connector Powered by Teradata, configure the -Dsqoop.connection.factories argument as follows:
  `-Dsqoop.connection.factories=com.cloudera.connector.teradata.TeradataManagerFactory`
- To use the Hortonworks Connector for Teradata (powered by the Teradata Connector for Hadoop), configure the -Dsqoop.connection.factories argument as follows:
  `-Dsqoop.connection.factories=org.apache.sqoop.teradata.TeradataManagerFactory`

Note: You do not need to configure the -Dsqoop.connection.factories argument if you run the mapping on the Spark engine.

For a complete list of the Sqoop arguments that you can configure, see the Sqoop documentation.

Creating a Connection to Access Sources or Targets

Create an HBase, HDFS, Hive, or JDBC connection before you import data objects, preview data, and profile data.

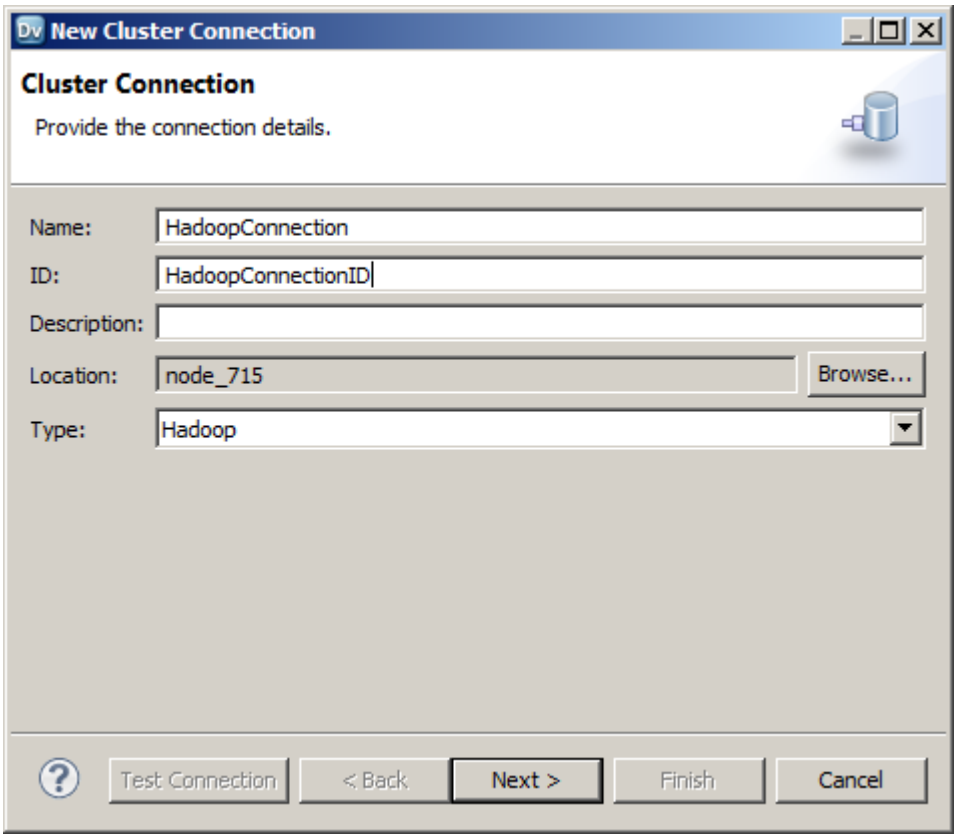
1. Click **Window** > **Preferences**.
2. Select **Informatica** > **Connections**.
3. Expand the domain in the **Available Connections** list.
4. Select the type of connection that you want to create:
   - To select an HBase connection, select **NoSQL** > **HBase**.
   - To select an HDFS connection, select **File Systems** > **Hadoop File System**.
   - To select a Hive connection, select **Database** > **Hive**.
   - To select a JDBC connection, select **Database** > **JDBC**.
5. Click **Add**.
6. Enter a connection name and optional description.
7. Click **Next**.
8. Configure the connection properties. For a Hive connection, you must choose the **Access Hive as a source or target** option to use Hive as a source or a target. The **Access Hive to run mappings in Hadoop cluster** options is no more applicable. To use the Hive driver to run mappings in the Hadoop cluster, use a Hadoop connection.
9. Click **Test Connection** to verify the connection.
10. Click **Finish**.

Creating a Hadoop Connection

Create a Hadoop connection before you run a mapping in the Hadoop environment.

1. Click **Window** > **Preferences**.
2. Select **Informatica** > **Connections**.
3. Expand the domain in the **Available Connections** list.
4. Select the **Cluster** connection type in the **Available Connections** list and click **Add**.

   The **New Cluster Connection** dialog box appears.
5. Enter the general properties for the connection.



6. Click **Next**.
7. Enter the Hadoop cluster properties and the common properties for the Hadoop connection.
8. Click **Next**.
9. Enter the Hive pushdown configuration properties and the Hive configuration.
10. Click **Next**.
11. If you are using the Blaze engine, enter the properties for the Blaze engine.
12. If you are using the Spark engine, enter the properties for the Spark engine.
13. Click **Finish**.

APPENDIX B

Multiple Blaze Instances on a Cluster

This appendix includes the following topics:

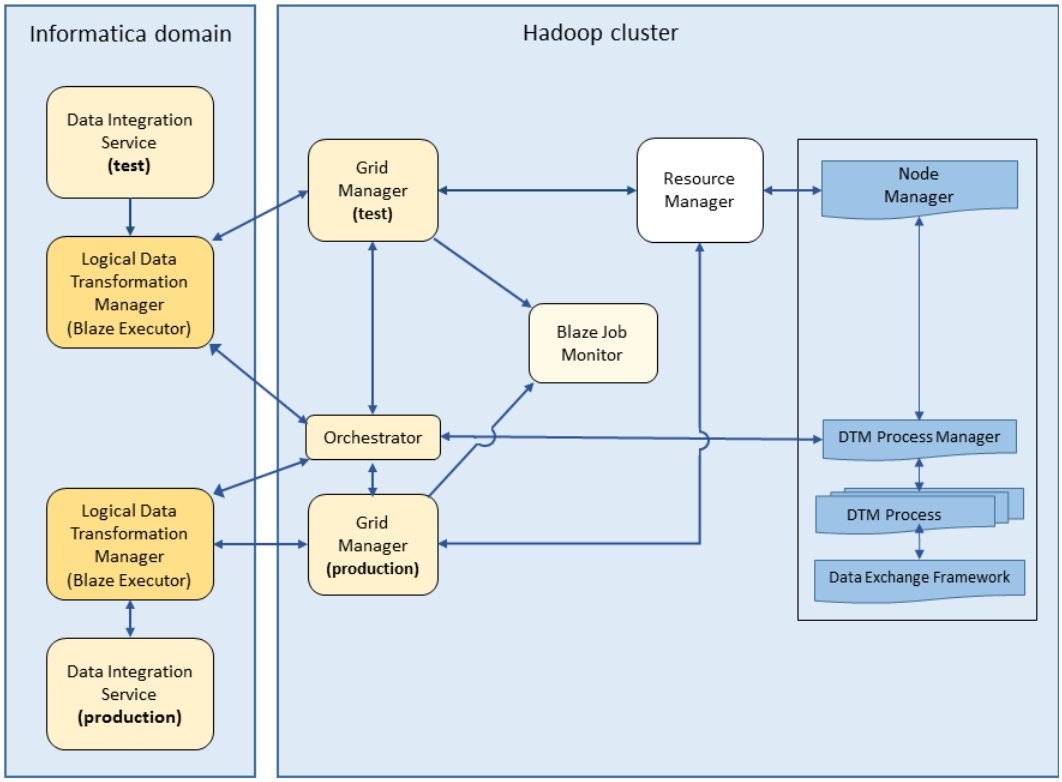
Overview

When you use the Blaze engine to run mappings, Blaze uses a Grid Manager at run time to allot tasks to various nodes in a Hadoop cluster. The Grid Manager aids in resource allocation.

You can use the same Hadoop cluster to stage your test environment and establish a production environment. To control resource use on the cluster, you can establish a separate Blaze instance for testing and another for production.

Each instance requires a separate Grid Manager. You create an additional Grid Manager by performing a series of steps to create separate infrastructure for each Blaze instance, including a unique namespace and a Hadoop connection for each Blaze instance to use.

The following image shows how a separate Data Integration Service on the domain creates a separate Grid Manager on the cluster:



The image shows how separate Data Integration Services use separate Blaze instances. Each instance uses a separate Grid Manager to communicate with the cluster resource manager to balance resources.

Perform the following steps to set up separate Blaze instances:

Step 1. Prepare the Hadoop cluster for the Blaze engine.

Step 2. Configure Data Integration Service properties.

Step 3. Update `hadoopEnv.properties`.

Step 4. Create a new Hadoop connection.

Step 5. Set mapping preferences.

Step 1. Prepare the Hadoop Cluster for the Blaze Engine

To run mappings on the Blaze engine, perform the following tasks:

1. Create an account for the Blaze engine user.
2. Create Blaze engine directories and grant permissions.
3. Grant permissions on the Hive source database.

Create a Blaze User Account

On all nodes in the Hadoop cluster, create an operating system user account for the user you want to run the additional Blaze instance. For example, run the following command:

```
useradd testuser1
```

Create Blaze Engine Directories and Grant Permissions

Create the following directories on the Hadoop cluster:

Local services log directory

Create a local services log directory on all nodes in the cluster and grant permissions to the Blaze user account. The hadoopEnv.properties file on the domain contains an entry for this directory. The file uses an environment variable, $HADOOP_NODE_INFA_HOME, that gets set to the Big Data Management installation directory. The default installation directory is `/opt/Informatica`. For example, run the following commands:

```
hadoop fs mdkir -p /opt/Informatica/blazeLogs
hadoop fs -chmod 777 /opt/Informatica/blazeLogs
```

If you use a different directory name, you must update the following property in the hadoopEnv.properties file: `infagrid.node.local.root.log.dir`

HDFS temporary working directory

Create a working directory on HDFS for the Blaze engine and grant permissions to the Blaze user account. For example, run the following commands:

```
hadoop fs mkdir -p /blaze/workdir
hadoop fs -chmod 777 /blaze/workdir
```

When you configure connection properties, you provide the path to this directory. Alternatively, you can create this directory when you create the connection.

Note: This directory is separate from the aggregated persistent log directory.

Verify that a persistent aggregated HDFS log directory exists on the cluster. For example, `/var/log/Hadoop-yarn/apps/Informatica`.

Note: It is not necessary to create a new directory for persistent logs. Both Blaze instances can use the same persistent aggregated HDFS log directory.

Grant Permissions on the Hive Source Database

Grant the Blaze user account CREATE TABLE permission on the Hive source database. The CREATE TABLE permission is required in the following situations:

- The Hive source table uses SQL standard-based authorization.
- A mapping contains a Lookup transformation with an SQL override.

Step 2. Configure Data Integration Service Properties

Configure Data Integration Service properties to enable two Blaze instances on the Hadoop environment.

You can create a Data Integration Service, or configure one that has not run mappings using the Blaze engine.

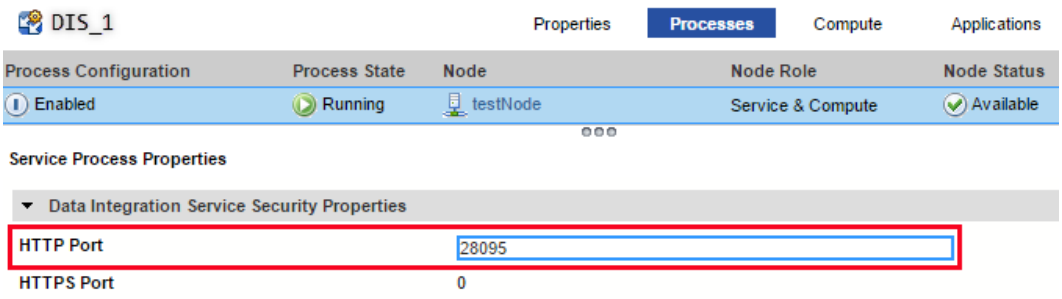
Configure Data Integration Service properties in the Administrator tool.

Data Integration Service Process Properties

Configure the following property on the **Processes** tab:

Property	Description
HTTP Port	The port that the Data Integration Service uses to communicate with the cluster over HTTP. Configure the port with a number that no other process uses.

The following image shows the HTTP Port property:



Data Integration Service Properties

The following table describes the Hadoop properties to configure for the Data Integration Service:

Property	Description
Informatica Home Directory on Hadoop	The Big Data Management home directory on every data node created by the installer. Default is `/opt/Informatica`.
Hadoop Distribution Directory	The directory containing a collection of Hive and Hadoop .jar files on the cluster from the RPM Install locations. The directory contains the minimum set of .jar files required to process Informatica mappings in a Hadoop environment. You can duplicate an existing Hadoop distribution folder if you want to use the same domain as an existing Grid Manager. For example, if you use the distribution folder `cloudera_cdh<version>` for production, you can duplicate it and name the folder like `cloudera_cdh<version>_test`. The distribution folders reside in `<Informatica installation directory>/services/shared/hadoop/`. For example, `/<Informatica installation directory>/services/shared/hadoop/cloudera_cdh<version>`. **Note:** When you use two Informatica domains to run mappings on the same cluster, do not duplicate the Hadoop distribution folder.
Data Integration Service Hadoop Distribution Directory	The Hadoop distribution directory on the Data Integration Service node. Type `/<Informatica installation directory>/Informatica/services/shared/hadoop/<Hadoop distribution name>_<version number>`. For example: `../../services/shared/hadoop/amazon_emr_5.0.0` Configure the Data Integration Service Hadoop Distribution Directory property with the folder on the domain that contains the Hadoop distribution settings that you want to use for the new Blaze instance. **Note:** The contents of the Data Integration Service Hadoop distribution directory must be identical to Hadoop distribution directories on the data nodes.

Step 3. Update hadoopEnv.properties

Update the `hadoopEnv.properties` file on each node where the Data Integration Service runs to configure an additional Blaze instance.

Open `hadoopEnv.properties` and back it up before you configure it. You can find the `hadoopEnv.properties` file in the following location:

```
<Informatica installation directory>/services/shared/hadoop/<distribution name>_<version number>/infaConf
```

Optionally Create a New Namespace

When the machine where the Data Integration Service runs contains two domains running on the same version of Informatica, you configure a new Blaze instance on the domain where you want to run the new Blaze instance.

In the "Advanced Configuration" section of `hadoopEnv.properties`, type the following property to designate a namespace for the Data Integration Service.

infagrid.cadi.namespace

Namespace for the Data Integration Service to use.

Configure the property as follows:

```
infagrid.cadi.namespace=<unique value>
```

For example,

```
infagrid.cadi.namespace=TestUser1_namespace
```

Configure Ports

Search for the following properties and enter port numbers that no other cluster processes use.

infagrid.blaze.console.jsfport

JSF port for the Blaze engine console.

Configure the property as follows:

```
infagrid.blaze.console.jsfport=<unique value>
```

For example,

```
infagrid.blaze.console.jsfport=9090
```

infagrid.blaze.console.httpport

HTTP port for the Blaze engine console.

Configure the property as follows:

```
infagrid.blaze.console.httpport=<unique value>
```

For example,

```
infagrid.blaze.console.httpport=9091
```

Configure Directory Paths

Search for the following properties and enter paths for the Blaze service logs and persistent logs.

infagrid.node.local.root.log.dir

Path for the Blaze service logs.

Note: This is the path that you configured in Step 1 as the local services log directory.

Configure the property as follows:

```
infagrid.node.local.root.log.dir=<directory path>
```

For example,

```
infagrid.node.local.root.log.dir=/opt/Informatica/blazeLogs
```

infacal.hadoop.logs.directory

Path in HDFS for the persistent Blaze logs.

Note: This is the path that you configured in Step 1 as the persistent log directory.

Configure the property as follows:

```
infacal.hadoop.logs.directory=<directory path>
```

For example,

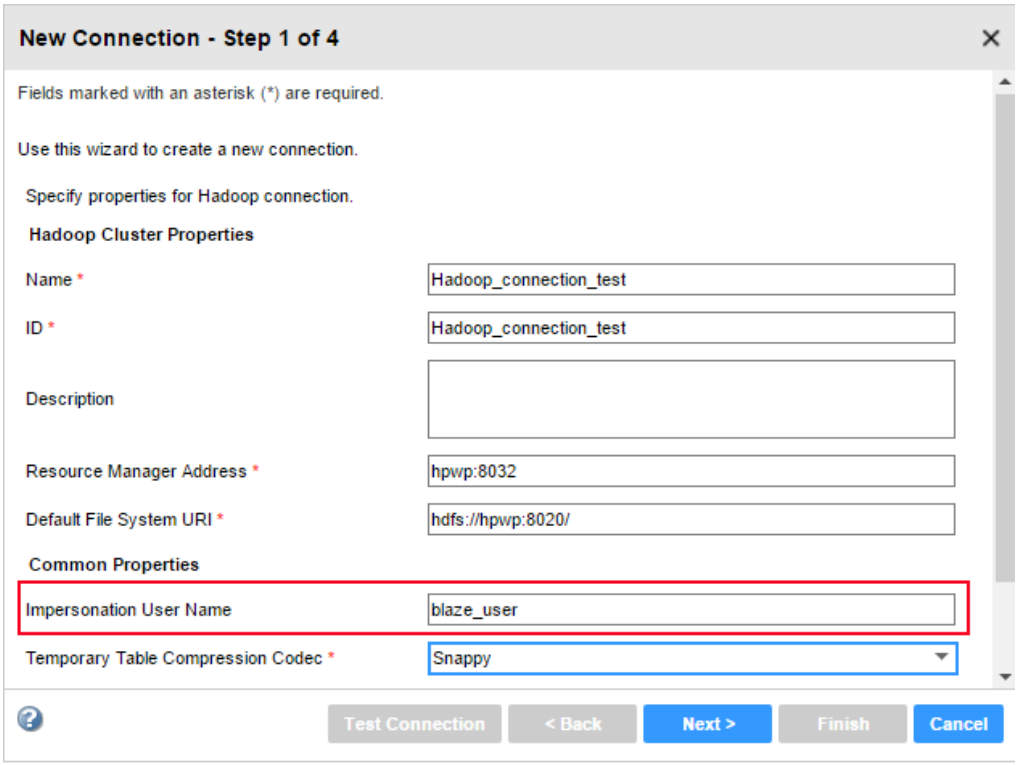
```
infacal.hadoop.logs.directory=infacal.hadoop.logs.directory=/var/log/Hadoop-yarn/
apps/Informatica
```

Step 4. Create a Hadoop Connection

Create a Hadoop connection for the Blaze instance to use.

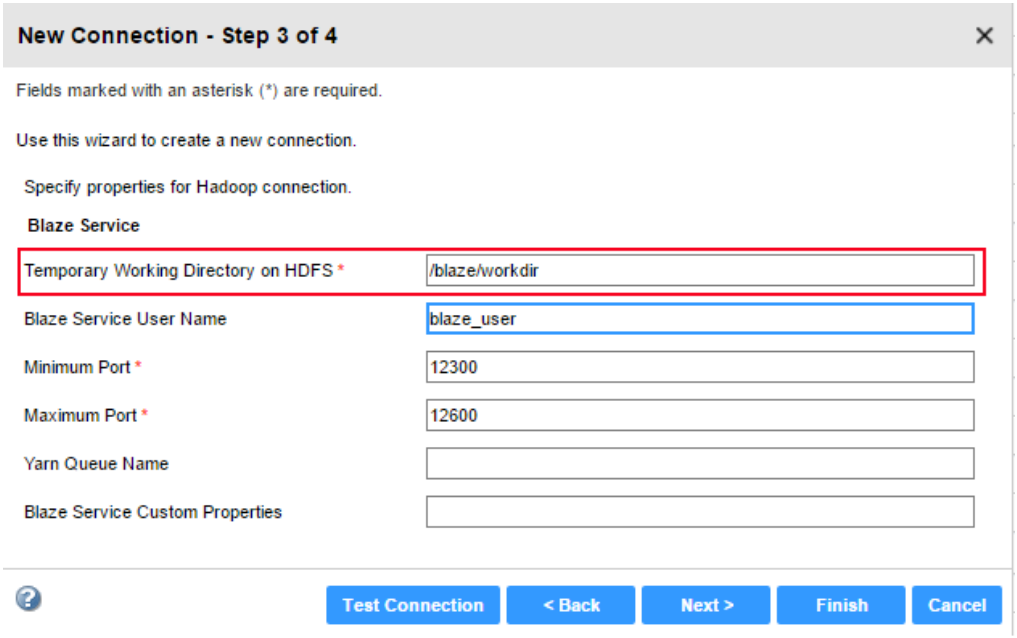
1. In Step 1 of the New Connection wizard, configure the Impersonation User Name property with the same impersonation user that you configured in Step 1, "Create a Blaze User Account."

   The following image shows the Impersonation User Name property in the New Connection wizard:



2. In Step 3 of the New Connection wizard, configure the Temporary Working Directory on HDFS property with the path that you configured on the cluster in "Create Blaze Engine Directories and Grant Permissions."

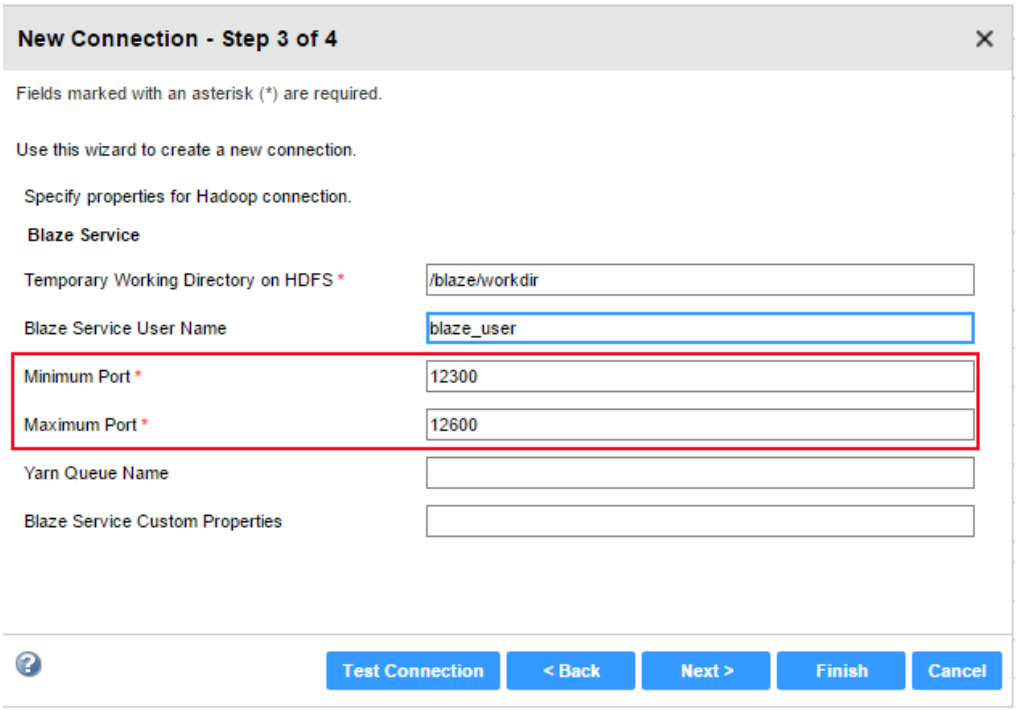
   The following image shows the Temporary Working Directory on HDFS property in the New Connection wizard:



3. Configure the Blaze Service User Name property with the same user name that you used to configure the Impersonation User in Step 1 of this topic.
4. Configure the Minimum Port and Maximum Port properties with a port range for the connection.

   You can supply a range within the port range of an existing Grid Manager, as long as ports are available when the mapping runs. The default range is 300 ports.

   The following image shows the Minimum Port and Maximum Port properties in the New Connection wizard:

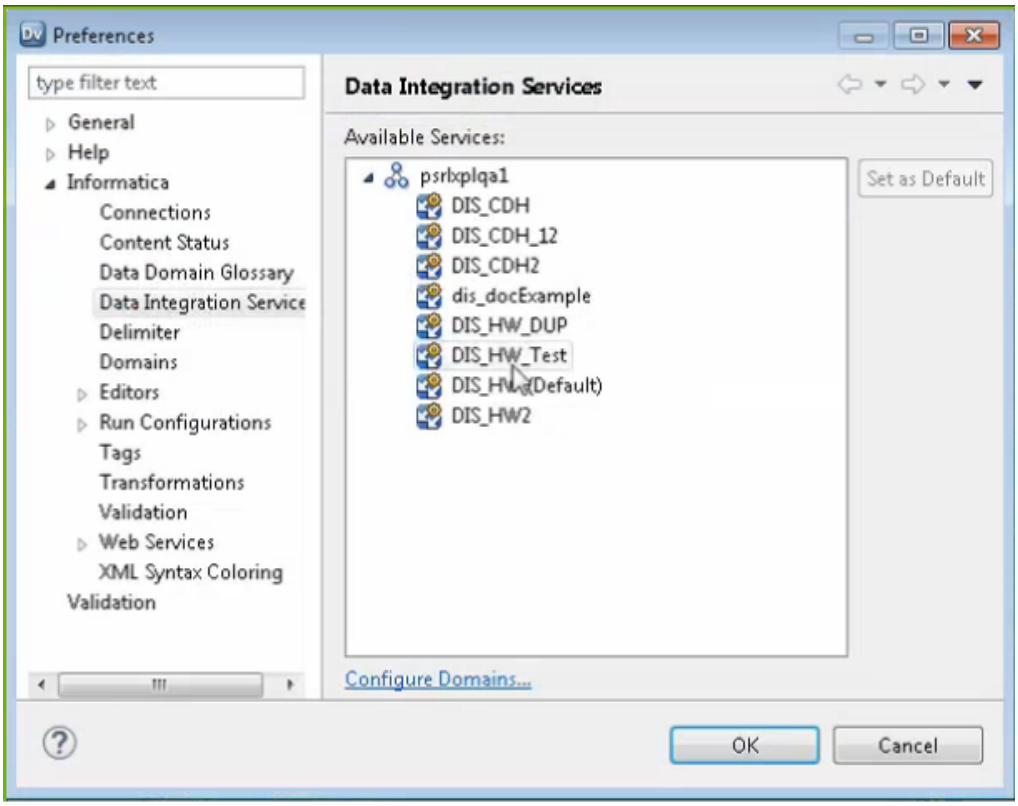


Step 5. Set Mapping Preferences

Before you run the mapping in the Developer tool, configure the mapping to use the Data Integration Service and Hadoop connection you want to use to run the mapping.

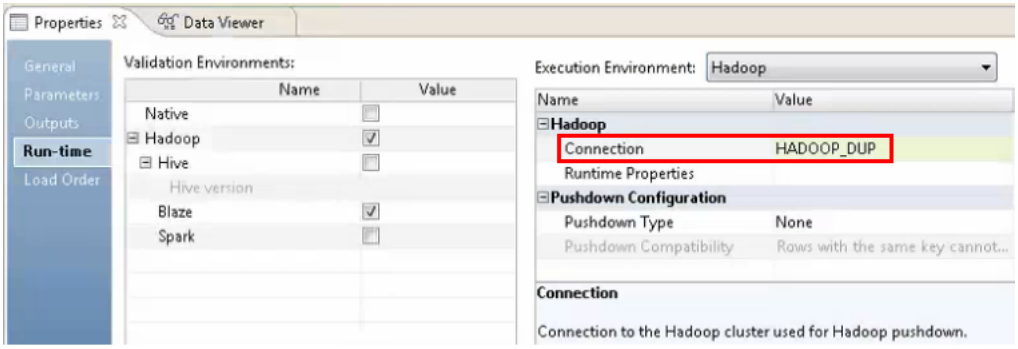
1. In the Developer tool, select **Mapping** > **Preferences**.
2. Expand the **Informatica** node, and then select **Data Integration Service**.

   The following image shows the list of available services in the **Preferences** window:



3. Select the Data Integration Service that you want to use, and then click **OK**.
4. In the **Properties** tab of the mapping, select the **Run-time** sub-tab.
5. In the Execution Environment area, set the Connection property to the Hadoop connection that you created.

   The following image shows the Connection property in the **Properties** tab:

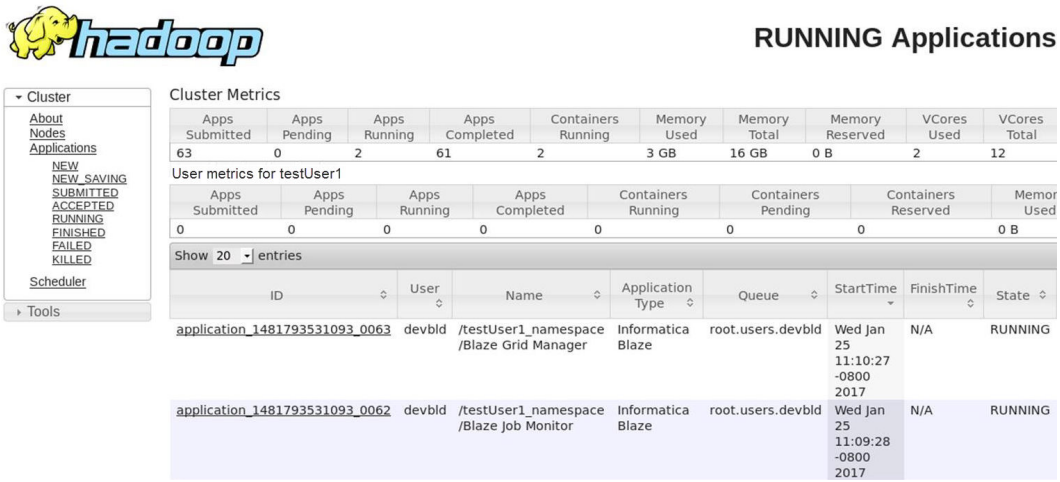


Result

The Data Integration Service creates a Grid Manager on the cluster the first time that it runs a mapping using the Blaze engine.

After you run the mapping, you can verify that the mapping used the Data Integration Service and new Grid Manager that you intended to use to run the mapping. Verify the resources that the mapping used by examining the Running Applications list in the Hadoop Resource Manager web interface. Look for applications that correspond to the namespace that you configured for the Blaze instance.

The following image shows applications with a name that includes the namespace, "testuser1_namespace," that you configured for the Grid Manager:



After completion of the mapping run, the Grid Manager persists. The mapping uses the same Grid Manager whenever it runs with the unique combination of Data Integration Service and connection.

To use multiple connections that use the same Grid Manager, use an identical namespace in each connection to refer to the Blaze instance. Verify that each connection also uses identical values for the Blaze user name and queue name. If you use different values for the Blaze user name and queue name to connect to the same Blaze instance, the mapping fails.

APPENDIX C

Configure Access to an SSL-Enabled Cluster

This appendix includes the following topics:

Configuring Big Data Management to Access an SSL-Enabled Cluster

When you configure the Informatica domain to communicate with an SSL-enabled cluster, the Developer tool client can import metadata from sources on the cluster, and the Data Integration Service can run mapping jobs on the cluster.

To access an SSL-enabled cluster, configure the Hive connection object to enable SSL and import security certificates to clients. Then configure the Data Integration Service properties with the location of SSL truststore files on the cluster. The cluster might use different locations for truststore files, depending on the resource location. For example, a cluster might store truststore files in separate locations for HDFS and Hive data, or it might store the files in one location.

Step 1. Configure the Connection String

If you created the Hive connection when you created cluster configurations, the cluster configuration creation wizard enables access to an SSL-enabled cluster.

If the Hive connection was not created by the cluster configuration wizard, add the following property-value pair to the connection string to each Hive connection object that accesses an SSL-enabled cluster:

```
ssl=true
```

For example:

```
jdbc:hive2://<hostname>:<port>/<db>;ssl=true
```

Note: Insert the `ssl=true` flag before the kerberos principal element when you create the Hive connection manually.

Step 2. Import Security Certificates

When you use custom, special, or self-signed security certificates to secure the Hadoop cluster, Informatica clients that connect to the cluster require these certificates to be present in the client machine truststore.

To connect to the Hadoop cluster to develop a mapping, the Developer tool requires security certificate aliases on the machine that hosts the Developer tool. To run a mapping, the machine that hosts the Data Integration Service requires these same certificate alias files.

Perform the following steps from the Developer tool host machine and from the Data Integration Service host machine:

1. Run the following command to export the certificates from the cluster:

   ```
   keytool -export -alias <alias name> -keystore <custom.truststore file location> -
   file <exported certificate file location> -storepass <password>
   ```

   For example:

   ```
   <java home>/jre/bin/keytool -export -alias <alias name> -keystore ~/
   custom.truststore -file ~/exported.cer
   ```

   The command produces a certificate file.

2. Choose to import security certificates to an SSL-enabled domain or a domain that is not SSL-enabled.
   - If the domain is SSL-enabled, then import the certificate alias file to the following locations:
     - The following path on the Developer tool machine: `<Developer Tool home>\clients\shared\security\infa_truststore.jks`
     - The following path on the machine that hosts the Data Integration Service: `<Informatica installation home>/services/shared/security/infa_truststore.jks`
   - If the Informatica domain is not SSL-enabled, import the security certificate alias file to the following locations:
     - The following path on the Developer tool machine: `<Developer Tool home>\clients\java\jre\lib\security\cacerts`
     - The following path on the machine that hosts the Data Integration Service: `<Informatica installation home>/java/jre/lib/security/cacerts`

INDEX

G

H

I

J

K

L

O

P

R

S

T

U