Informatica Big Data Trial Sandbox for Hortonworks (Version 9.6.1)

User Guide
This product includes software licensed under the Academic Free License (http://www.opensource.org/licenses/afl-3.0.php), the Common Development and Distribution License (http://www.opensource.org/licenses/cddl1.php) the Common Public License (http://www.opensource.org/licenses/cpl1.0.php), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (http://www.opensource.org/licenses/b sd-license.php), the new BSD License (http://opensource.org/licenses/BSD-3-Clause), the MIT License (http://www.opensource.org/licenses/mit-license.php), the Artistic License (http://www.opensource.org/licenses/artistic-license-1.0) and the Initial Developer's Public License Version 1.0 (http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/).

This product includes software copyright © 2003-2006 Joe Walnes, 2006-2007 XStream Committers. All rights reserved. Permissions and limitations regarding this software are subject to terms available at http://xstream.codehaus.org/license.html. This product includes software developed by the Indiana University Extreme! Lab. For further information please visit http://www.extreme.indiana.edu/

This product includes software Copyright (c) 2013 Frank Balluffi and Markus Moeller. All rights reserved. Permissions and limitations regarding this software are subject to terms of the MIT license.

This Software is protected by U.S. Patent Numbers 5,794,246; 6,014,670; 6,016,501; 6,029,178; 6,032,158; 6,035,307; 6,044,374; 6,092,086; 6,208,990; 6,339,775; 6,640,226; 6,789,096; 6,823,373; 6,850,947; 6,895,471; 7,117,215; 7,162,643; 7,243,110; 7,254,590; 7,281,001; 7,421,458; 7,496,888; 7,523,121; 7,584,422; 7,676,516; 7,720,842; 7,721,270; 7,774,791; 8,065,296; 8,150,803; 8,166,048; 8,166,071; 8,200,622; 8,224,973; 8,271,477; 8,327,419; 8,386,435; 8,392,440; 8,453,159; 8,458,230; and RE44,478, International Patents and other Patents Pending.

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

NOTICES

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

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.

2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

Part Number: IN-BDETrial-961-000-0001
# Table of Contents

**Preface** \[iv\]
- Informatica Resources. \[iv\]
- Informatica My Support Portal. \[iv\]
- Informatica Documentation. \[iv\]
- Informatica Web Site. \[iv\]
- Informatica How-To Library. \[iv\]
- Informatica Knowledge Base. \[v\]
- Informatica Support YouTube Channel. \[v\]
- Informatica Marketplace. \[v\]
- Informatica Velocity. \[v\]
- Informatica Global Customer Support. \[v\]

**Chapter 1: Introduction to Big Data Trial Sandbox for Hortonworks** \[1\]
- Big Data Trial Sandbox for Hortonworks Overview. \[1\]
- Big Data Access. \[2\]
- High Performance Processing in the Native Environment. \[2\]
- High Performance Processing in a Hive Environment. \[3\]
  - Hive Environment Processing Architecture. \[3\]
- Big Data Processing Example. \[4\]

**Chapter 2: Connections** \[6\]
- Connections Overview. \[6\]
- HDFS Connection Properties. \[6\]
- Hive Connection Properties. \[7\]
- Creating a Connection. \[13\]

**Chapter 3: Mappings in the Native Environment** \[14\]
- Mappings in the Native Environment Overview. \[14\]
- Data Processor Mappings. \[15\]
- HDFS Mappings. \[15\]
  - HDFS Data Extraction Mapping Example. \[15\]
- Hive Mappings. \[16\]
  - Hive Mapping Example. \[17\]
- Social Media Mappings. \[17\]
  - Twitter Mapping Example. \[18\]

**Chapter 4: Mappings in a Hive Environment** \[19\]
- Mappings in a Hive Environment Overview. \[19\]
- Sources in a Hive Environment. \[20\]
  - Flat File Sources. \[20\]
Hive Sources .................................................... 21
Relational Sources ............................................ 21
Targets in a Hive Environment ......................... 21
Flat File Targets .............................................. 22
HDFS Flat File Target ..................................... 22
Hive Targets ................................................ 22
Relational Targets ........................................ 22
Transformations in a Hive Environment ............... 23
Functions in a Hive Environment ..................... 25
Mappings in a Hive Environment ..................... 26
Datatypes in a Hive Environment ..................... 26
Variable Ports in a Hive Environment ............... 27
Workflows in a Hive Environment ..................... 27
Configuring a Mapping to Run in a Hive Environment ............................................... 27
Hive Execution Plan ........................................ 28
Hive Execution Plan Details ............................. 28
Viewing the Hive Execution Plan for a Mapping ... 28
Monitoring a Mapping ....................................... 28
Logs ........................................................... 29
Troubleshooting a Mapping in a Hive Environment ............................................... 29

Chapter 5: Profiles ............................................. 30
Profiles Overview ........................................... 30
Native and Hadoop Environments ....................... 31
Supported Data Source and Run-time Environments ............................................... 31
Run-time Environment Setup and Validation ........ 32
Run-time Environment and Profile Performance .... 33
Profile Types on Hadoop .................................... 33
Column Profiles on Hadoop .............................. 33
Rule Profiles on Hadoop .................................... 34
Data Domain Discovery on Hadoop .................... 34
Running a Single Data Object Profile on Hadoop .................... 34
Running Multiple Data Object Profiles on Hadoop .................... 35
Monitoring a Profile ........................................ 35
Viewing Profile Results .................................... 36
Troubleshooting ............................................. 36

Chapter 6: Native Environment Optimization .............. 38
Native Environment Optimization Overview ........ 38
Processing Big Data on Partitions ..................... 38
Partitioned Model Repository Mappings .............. 38
Partition Optimization .................................... 39
High Availability .......................................... 39
Preface

The Big Data Trial Sandbox for Hortonworks User Guide provides information about how to configure Informatica products for Hadoop.

Informatica Resources

Informatica My Support Portal


The site contains product information, user group information, newsletters, access to the Informatica customer support case management system (ATLAS), the Informatica How-To Library, the Informatica Knowledge Base, Informatica Product Documentation, and access to the Informatica user community.

Informatica Documentation

The Informatica Documentation team takes every effort to create accurate, usable documentation. If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com. We will use your feedback to improve our documentation. Let us know if we can contact you regarding your comments.

The Documentation team updates documentation as needed. To get the latest documentation for your product, navigate to Product Documentation from http://mysupport.informatica.com.

Informatica Web Site

You can access the Informatica corporate web site at http://www.informatica.com. The site contains information about Informatica, its background, upcoming events, and sales offices. You will also find product and partner information. The services area of the site includes important information about technical support, training and education, and implementation services.

Informatica How-To Library

As an Informatica customer, you can access the Informatica How-To Library at http://mysupport.informatica.com. The How-To Library is a collection of resources to help you learn more about Informatica products and features. It includes articles and interactive demonstrations that provide solutions to common problems, compare features and behaviors, and guide you through performing specific real-world tasks.
Informatica Knowledge Base

As an Informatica customer, you can access the Informatica Knowledge Base at http://mysupport.informatica.com. Use the Knowledge Base to search for documented solutions to known technical issues about Informatica products. You can also find answers to frequently asked questions, technical white papers, and technical tips. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team through email at KB.Feedback@informatica.com.

Informatica Support YouTube Channel

You can access the Informatica Support YouTube channel at http://www.youtube.com/user/INFASupport. The Informatica Support YouTube channel includes videos about solutions that guide you through performing specific tasks. If you have questions, comments, or ideas about the Informatica Support YouTube channel, contact the Support YouTube team through email at supportvideos@informatica.com or send a tweet to @INFASupport.

Informatica Marketplace

The Informatica Marketplace is a forum where developers and partners can share solutions that augment, extend, or enhance data integration implementations. By leveraging any of the hundreds of solutions available on the Marketplace, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at http://www.informaticamarketplace.com.

Informatica Velocity

You can access Informatica Velocity at http://mysupport.informatica.com. 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 have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Global Customer Support

You can contact a Customer Support Center by telephone or through the Online Support.

Online Support requires a user name and password. You can request a user name and password at http://mysupport.informatica.com.

Introduction to Big Data Trial Sandbox for Hortonworks

This chapter includes the following topics:

- Big Data Trial Sandbox for Hortonworks Overview, 1
- Big Data Access, 2
- High Performance Processing in the Native Environment, 2
- High Performance Processing in a Hive Environment, 3
- Big Data Processing Example, 4

Big Data Trial Sandbox for Hortonworks Overview

The Big Data Trial Sandbox for Hortonworks includes basic functionality associated with Informatica products to run mappings in a native and Hive environment.

You can use the following functionality associated with big data:

Access big data sources
Access unstructured and semi-structured data, social media data, and data in Hive and HDFS.

Replicate data
Replicate large amounts of transactional data between heterogeneous databases and platforms.

Configure high-performance processing in the native environment
Enable partitioning to process partitions of data in parallel, and process data through highly available application services in the domain.

Configure high-performance processing in a Hive environment
Distribute mapping and profile processing across cluster nodes in a Hive environment.

You can process data in the native environment or a Hive environment. You can run Model repository mappings and profiles on the Data Integration Service. In the native environment, the Data Integration Service processes the data. In a Hive environment, nodes in a Hadoop cluster process the data.
Big Data Access

In addition to relational and flat file data, you can access unstructured and semi-structured data, social media data, and data in a Hive or Hadoop Distributed File System (HDFS) environment.

You can access the following types of data:

**Transaction data**

You can access different types of transaction data, including data from relational database management systems, online transaction processing systems, online analytical processing systems, enterprise resource planning systems, customer relationship management systems, mainframe, and cloud.

**Unstructured and semi-structured data**

You can use parser transformations to read and transform unstructured and semi-structured data. For example, you can use the Data Processor transformation in a workflow to parse a Microsoft Word file to load customer and order data into relational database tables.

You can use HParse to transform complex data into flattened, usable formats for Hive, PIG, and MapReduce processing. HParse processes complex files, such as messaging formats, HTML pages and PDF documents. HParse also transforms formats such as ACORD, HIPAA, HL7, EDI-X12, EDIFACT, AFP, and SWIFT.

**Social media data**

You can use PowerExchange adapters for social media to read data from social media web sites like Facebook, Twitter, and LinkedIn. You can also use the PowerExchange for DataSift to extract real-time data from different social media web sites and capture data from DataSift regarding sentiment and language analysis. You can use PowerExchange for Web Content-Kapow to extract data from any web site.

**Data in Hive and HDFS**

You can use other PowerExchange adapters to read data from or write data to Hadoop. For example, you can use PowerExchange for Hive to read data from or write data to Hive. Also, you can use PowerExchange for HDFS to extract data from and load data to HDFS.

High Performance Processing in the Native Environment

You can optimize the native environment to process big data fast and reliably. You can process partitions of a session or mapping in parallel. You can also enable high availability.

You can enable the following features to optimize the native environment:

**Partitioning for Model repository mappings**

You can enable the Data Integration Service process to maximize parallelism when it runs mappings. When you maximize parallelism, the Data Integration Service dynamically divides the underlying data into partitions. When the Data Integration Service adds partitions, it increases the number of processing threads, which can increase mapping performance. The Data Integration Service performs the extract, transformation, and load for each partition in parallel.
High availability

You can enable high availability to eliminate single points of failure for domain, application services, and application clients. The domain, application services, and application clients can continue running despite temporary network or hardware failures.

For example, if you run the Data Integration Service on a single node and you enable high availability, you can configure backup nodes in case the primary node becomes unavailable. If the node running master service process becomes unavailable while running a workflow, the Data Integration Service can recover the workflow based on the workflow state and recovery strategy. If the workflow was enabled for high availability recovery, the Data Integration Service restores the state of operation for the workflow and recovers the workflow from the point of interruption.

If the node running the worker service process becomes unavailable while running tasks of a workflow, the master service process can recover tasks based on task state and recovery strategy.

High Performance Processing in a Hive Environment

You can run Model repository mappings and profiles in a Hive environment to process large amounts of data of 10 terabytes or more. In the Hive environment, the Data Integration Service converts the mapping or profile into MapReduce programs to enable the Hadoop cluster to process the data.

Hive Environment Processing Architecture

You can run Model repository mappings or profiles in a Hive environment.

To run a mapping or profile in a Hive environment, 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 MapReduce jobs to the Hadoop cluster:

1. The Hive driver sends the MapReduce jobs to the Job Tracker in the Hive environment.
2. The JobTracker retrieves a list of TaskTracker nodes that can process the MapReduce jobs from the NameNode.
3. The JobTracker assigns MapReduce jobs to TaskTracker 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.

Big Data Processing Example

Every week, an investment banking organization manually calculates the popularity and risk of stocks, and then matches stocks to each customer based on the preferences of the customer. However, the organization now wants you to automate this process.

You use the Developer tool to create a workflow that calculates the popularity and risk of each stock, matches stocks to each customer, and then sends an email with a list of stock recommendations for all customers. To determine the popularity of a stock, you count the number of times that the stock was included in Twitter feeds and the number of times customers inquired about the stock on the company stock trade web site.

The following diagram shows the components of the workflow:

You configure the workflow to complete the following tasks:
1. **Extract and count the number of inquiries about stocks from weblogs.**
   
   Extracts the inquiries about each stock from the weblogs, and then counts the number of inquiries about each stock. The weblogs are from the company stock trade web site.

2. **Extract and count the number of tweets for each stock from Twitter.**
   
   Extracts tweets from Twitter, and then counts the number of tweets about each stock.

3. **Extract market data and calculate the risk of each stock based on market data.**
   
   Extracts the daily high stock value, daily low stock value, and volatility of each stock from a flat file provided by a third-party vendor. The workflow calculates the risk of each stock based on the extracted market data.

4. **Combine the inquiry count, tweet count, and risk for each stock.**
   
   Combines the inquiry count, tweet count, and risk for each stock from the weblogs, Twitter, and market data, respectively.

5. **Extract historical stock transactions for each customer.**
   
   Extracts historical stock purchases of each customer from a database.

6. **Calculate the average risk and average popularity of the stocks purchased by each customer.**
   
   Calculates the average risk and average popularity of all stocks purchased by each customer.

7. **Match stocks to each customer based on their preferences.**
   
   Matches stocks that have the same popularity and risk as the average popularity and average risk of the stocks that the customer previously purchased.

8. **Load stock recommendations into the data warehouse.**
   
   Loads the stock recommendations into data warehouse to retain a history of the recommendations.

9. **Send an email with stock recommendations.**
   
   Consolidates the stock recommendations for all customers, and sends an email with the list of recommendations.

After you create the workflow, you configure it to run in a Hive environment because the workflow must process 15 terabytes of data each time it creates recommendations for customers.
This chapter includes the following topics:

- Connections Overview, 6
- HDFS Connection Properties, 6
- Hive Connection Properties, 7
- Creating a Connection, 13

Connections Overview

Define the connections you want to use to access data in Hive or HDFS.

You can create the following types of connections:

- HDFS connection. Create an HDFS connection to read data from or write data to the Hadoop cluster.
- Hive connection. Create a Hive connection to access Hive data or run Informatica mappings in the Hadoop cluster. Create a Hive connection in the following connection modes:
  - Use the Hive connection to access Hive as a source or target. If you want to use Hive as a target, you need to have the same connection or another Hive connection that is enabled to run mappings in the Hadoop cluster. You can access Hive as a source if the mapping is enabled for the native or Hive environment. You can access Hive as a target only if the mapping is run in the Hadoop cluster.
  - Use the Hive connection to validate or run an Informatica mapping in the Hadoop cluster. Before you run mappings in the Hadoop cluster, review the information in this guide about rules and guidelines for mappings that you can run in the Hadoop cluster.

You can create the connections using the Developer tool, Administrator tool, and infacmd.

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.

HDFS Connection Properties

Use an HDFS connection to access files in the Hadoop Distributed File System.
The following table describes the properties for an HDFS connection:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>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. It cannot exceed 128 characters, contain spaces, or contain the following special characters: `~ ! $ % ^ * ( ) _ + = { [ ] }</td>
</tr>
<tr>
<td>ID</td>
<td>The 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.</td>
</tr>
<tr>
<td>Description</td>
<td>The description of the connection. The description cannot exceed 765 characters.</td>
</tr>
<tr>
<td>Location</td>
<td>The domain where you want to create the connection.</td>
</tr>
<tr>
<td>Type</td>
<td>The connection type. Default is Hadoop File System.</td>
</tr>
<tr>
<td>User Name</td>
<td>The user name to access HDFS.</td>
</tr>
</tbody>
</table>
| NameNode URI    | The URI to access HDFS. Use the following format to specify the NameNode URI in Cloudera and Hortonworks distributions:  
|                 | 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).                                                               |
|                 | Use one of the following formats to specify the NameNode URI in MapR distribution:                                                             |
|                 | - maprfs://                                                                                                                                  |
|                 | - maprfs://mapr/my.cluster.com/                                                                                                                 |
|                 | Where `my.cluster.com` is the cluster name that you specify in the `mapr-clusters.conf` file.                                                      |

**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 or the Developer tool.

**Note:** The order of the connection properties might vary depending on the tool where you view them.
The following table describes Hive connection properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>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: ~ ` ! $ % ^ &amp; * ( ) - + = { }</td>
</tr>
<tr>
<td>ID</td>
<td>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.</td>
</tr>
<tr>
<td>Description</td>
<td>The description of the connection. The description cannot exceed 4000 characters.</td>
</tr>
<tr>
<td>Location</td>
<td>The domain where you want to create the connection.</td>
</tr>
<tr>
<td>Type</td>
<td>The connection type. Select Hive.</td>
</tr>
</tbody>
</table>
| Connection Modes | Hive connection mode. Select at least one of the following options:  
- Access Hive as a source or target. Select this option if you want to use the connection to access the Hive data warehouse. If you want to use Hive as a target, you need to enable the same connection or another Hive connection to run mappings in the Hadoop cluster.  
- Use Hive to run mappings in Hadoop cluster. Select this option if you want to use the connection to run mappings in the Hadoop cluster.  
You can select both the options. Default is **Access Hive as a source or target**. |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>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 uses Kerberos authentication, the principal name for the JDBC connection string and the user name must match. Else, the user name depends on the behavior of the JDBC driver. 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.</td>
</tr>
<tr>
<td>Common Attributes to Both the Modes: Environment SQL</td>
<td>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 Hive metastore. If the Hive connection is used to run mappings in the 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 the 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 either 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 also use environment SQL to define Hadoop or Hive parameters that you intend to use in the PreSQL commands or in custom queries. If the Hive connection is used to run mappings in the Hadoop cluster, only the environment SQL of the Hive connection is executed. The different environment SQL commands for the connections of the Hive source or target are not executed, even if the Hive sources and targets are on different clusters.</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata Connection String</td>
<td>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:hive://&lt;hostname&gt;:&lt;port&gt;/&lt;db&gt; To connect to HiveServer2, specify the connection string in the following format: jdbc:hive2://&lt;hostname&gt;:&lt;port&gt;/&lt;db&gt; Where - hostname is name or IP address of the machine on which HiveServer or HiveServer2 runs. - port is the port number on which HiveServer or 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.</td>
</tr>
<tr>
<td>Bypass Hive JDBC Server</td>
<td>JDBC driver mode. Select the checkbox to use the embedded JDBC driver (embedded 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 in the Hadoop cluster. If you choose the non-embedded mode, you must configure the Data Access Connection String. The JDBC embedded mode is preferred to the non-embedded mode.</td>
</tr>
<tr>
<td>Data Access Connection String</td>
<td>The connection string used 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:hive://&lt;hostname&gt;:&lt;port&gt;/&lt;db&gt; To connect to HiveServer2, specify the non-embedded JDBC mode connection string in the following format: jdbc:hive2://&lt;hostname&gt;:&lt;port&gt;/&lt;db&gt; Where - hostname is name or IP address of the machine on which HiveServer or HiveServer2 runs. - port is the port number on which HiveServer or 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.</td>
</tr>
</tbody>
</table>
Properties to Run Mappings in 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 in the Hadoop cluster:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>Namespace for tables. Use the name default for tables that do not have a specified database name.</td>
</tr>
<tr>
<td>Default FS URI</td>
<td>The URI to access the default Hadoop Distributed File System. Use the following connection URI: hdfs://&lt;node name&gt;:&lt;port&gt; Where</td>
</tr>
<tr>
<td></td>
<td>- node name is the host name or IP address of the NameNode.</td>
</tr>
<tr>
<td></td>
<td>- port is the port on which the NameNode listens for remote procedure calls (RPC).</td>
</tr>
<tr>
<td>JobTracker/Yarn Resource Manager URI</td>
<td>The service within Hadoop that submits the MapReduce tasks to specific nodes in the cluster. Use the following format: &lt;hostname&gt;:&lt;port&gt; Where</td>
</tr>
<tr>
<td></td>
<td>- hostname is the host name or IP address of the JobTracker or Yarn resource manager.</td>
</tr>
<tr>
<td></td>
<td>- port is the port on which JobTracker or Yarn resource manager listens for remote procedure calls (RPC).</td>
</tr>
<tr>
<td>MapR distribution supports a highly available JobTracker. If you are using MapR distribution, define the JobTracker URI in the following format: maprfs://</td>
<td></td>
</tr>
<tr>
<td>Hive Warehouse Directory on HDFS</td>
<td>The absolute HDFS file path of the default database for the warehouse, which is local to the cluster. For example, the following file path specifies a local warehouse: /user/hive/warehouse</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Advanced Hive/Hadoop Properties      | Configures or overrides Hive or Hadoop cluster properties in hive-site.xml on the machine on which the Data Integration Service runs. You can specify multiple properties. Use the following format: `<property1>=<value>`  
Where  
- `property1` is a Hive or Hadoop property in hive-site.xml.  
- `value` is the value of the Hive or Hadoop property.  
To specify multiple properties use `&` 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 |
| Metastore Execution Mode             | Controls whether to connect to a remote metastore or a local metastore. By default, local is selected. For a local metastore, you must specify the Metastore Database URI, Driver, Username, and Password. For a remote metastore, you must specify only the Remote Metastore URI.                                                                                  |
| Metastore Database URI                | The JDBC connection URI used to access the data store in a local metastore setup. Use the following connection URI: jdbc:<datastore type>://<node name>:<port>/<database name>  
where  
- `node name` is the host name or IP address of the data store.  
- `data store type` is the type of the data store.  
- `port` is the port on which the data store listens for remote procedure calls (RPC).  
- `database name` is the name of the database.  
For example, the following URI specifies a local metastore that uses MySQL as a data store: jdbc:mysql://hostname23:3306/metastore                                                                                                                                 |
<p>| Metastore Database Driver            | Driver class name for the JDBC data store. For example, the following class name specifies a MySQL driver: com.mysql.jdbc.Driver                                                                                                                                                                                                                             |
| Metastore Database Username          | The metastore database user name.                                                                                                                                                                                                                                                                                                                                                      |</p>
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metastore Database Password</td>
<td>The password for the metastore user name.</td>
</tr>
<tr>
<td>Remote Metastore URI</td>
<td>The metastore URI used to access metadata in a remote metastore setup. For</td>
</tr>
<tr>
<td></td>
<td>a remote metastore, you must specify the Thrift server details. Use the</td>
</tr>
<tr>
<td></td>
<td>following connection URI:</td>
</tr>
<tr>
<td></td>
<td><code>thrift://&lt;hostname&gt;:&lt;port&gt;</code></td>
</tr>
<tr>
<td></td>
<td>Where</td>
</tr>
<tr>
<td></td>
<td>- <code>hostname</code> is name or IP address of the Thrift metastore server.</td>
</tr>
<tr>
<td></td>
<td>- <code>port</code> is the port on which the Thrift server is listening.</td>
</tr>
</tbody>
</table>

### Creating a Connection

Create a connection before you import data objects, preview data, profile data, and run mappings.

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 a Hive connection, select **Database > Hive**.
   - To select an HDFS connection, select **File Systems > Hadoop File System**.
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 Hive connection mode and specify the commands for environment SQL. The SQL commands apply to both the connection modes. Select at least one of the following connection modes:
   - **Access Hive as a source or target**
     - Use the connection to access Hive data. If you select this option and click **Next**, the **Properties to Access Hive as a source or target** page appears. Configure the connection strings.
   - **Run mappings in a Hadoop cluster**
     - Use the Hive connection to validate and run Informatica mappings in the Hadoop cluster. If you select this option and click **Next**, the **Properties used to Run Mappings in the Hadoop Cluster** page appears. Configure the properties.
9. Click **Test Connection** to verify the connection.
   - You can test a Hive connection that is configured to access Hive data. You cannot test a Hive connection that is configured to run Informatica mappings in the Hadoop cluster.
10. Click **Finish**.
Chapter 3

Mappings in the Native Environment

This chapter includes the following topics:

- Mappings in the Native Environment Overview, 14
- Data Processor Mappings, 15
- HDFS Mappings, 15
- Hive Mappings, 16
- Social Media Mappings, 17

Mappings in the Native Environment Overview

You can run a mapping in the native or Hive environment. In the native environment, the Data Integration Service runs the mapping from the Developer tool. You can run standalone mappings or mappings that are a part of a workflow.

In the native environment, you can read and process data from large unstructured and semi-structured files, Hive, or social media web sites. You can include the following objects in the mappings:

- Hive sources
- Flat file sources or targets in the local system or in HDFS
- Complex file sources in the local system or in HDFS
- Data Processor transformations to process unstructured and semi-structured file formats
- Social media sources

You can also import PowerCenter mappings in the Developer tool and run them in the native environment.
Data Processor Mappings

The Data Processor transformation processes unstructured and semi-structured file formats in a mapping. It converts source data to flat CSV records that MapReduce applications can process.

You can configure the Data Processor transformation to process messaging formats, HTML pages, XML, and PDF documents. You can also configure it to transform structured formats such as ACORD, HIPAA, HL7, EDI-X12, EDIFACT, AFP, and SWIFT.

For example, an application produces hundreds of data files per second and writes the files to a directory. You can create a mapping that extracts the files from the directory, passes them to a Data Processor transformation, and writes the data to a target.

HDFS Mappings

Create an HDFS mapping to read or write to HDFS.

You can read and write fixed-width and delimited file formats. You can read or write compressed files. You can read text files and binary file formats such as sequence file from HDFS. You can specify the compression format of the files. You can use the binary stream output of the complex file data object as input to a Data Processor transformation to parse the file.

You can define the following objects in an HDFS mapping:

- Flat file data object or complex file data object operation as the source to read data from HDFS.
- Transformations.
- Flat file data object as the target to write data to HDFS or any target.

Validate and run the mapping. You can deploy the mapping and run it or add the mapping to a Mapping task in a workflow.

HDFS Data Extraction Mapping Example

Your organization needs to analyze purchase order details such as customer ID, item codes, and item quantity. The purchase order details are stored in a semi-structured compressed XML file in HDFS. The hierarchical data includes a purchase order parent hierarchy level and a customer contact details child hierarchy level. Create a mapping that reads all the purchase records from the file in HDFS. The mapping must convert the hierarchical data to relational data and write it to a relational target.

You can use the extracted data for business analytics.

The following figure shows the example mapping:

You can use the following objects in the HDFS mapping:

HDFS Input

- Read_Complex_File

Data Processor Transaction

- Data_Processor_Transaction

- XML_to_Relational_Data_Object

The input, Read_Complex_File, is a compressed XML file stored in HDFS.
Data Processor Transformation

The Data Processor transformation, Data_Processor_XML_to_Relational, parses the XML file and provides a relational output.

Relational Output

The output, Write_Relational_Data_Object, is a table in an Oracle database.

When you run the mapping, the Data Integration Service reads the file in a binary stream and passes it to the Data Processor transformation. The Data Processor transformation parses the specified file and provides a relational output. The output is written to the relational target.

You can configure the mapping to run in a native or Hive run-time environment.

Complete the following tasks to configure the mapping:

1. Create an HDFS connection to read files from the Hadoop cluster.
2. Create a complex file data object read operation. Specify the following parameters:
   - The file as the resource in the data object.
   - The file compression format.
   - The HDFS file location.
3. Optionally, you can specify the input format that the Mapper uses to read the file.
4. Drag and drop the complex file data object read operation into a mapping.
5. Create a Data Processor transformation. Configure the following properties in the Data Processor transformation:
   - An input port set to buffer input and binary datatype.
   - Relational output ports depending on the number of columns you want in the relational output. Specify the port size for the ports. Use an XML schema reference that describes the XML hierarchy. Specify the normalized output that you want. For example, you can specify PurchaseOrderNumber_Key as a generated key that relates the Purchase Orders output group to a Customer Details group.
   - Create a Streamer object and specify Streamer as a startup component.
6. Create a relational connection to an Oracle database.
7. Import a relational data object.
8. Create a write transformation for the relational data object and add it to the mapping.

Hive Mappings

Based on the mapping environment, you can read data from or write data to Hive.

In a native environment, you can read data from Hive. To read data from Hive, complete the following steps:

1. Create a Hive connection.
2. Configure the Hive connection mode to access Hive as a source or target.
3. Use the Hive connection to create a data object to read from Hive.
4. Add the data object to a mapping and configure the mapping to run in the native environment.
You can write to Hive in a Hive environment. To write data to Hive, complete the following steps:

1. Create a Hive connection.
2. Configure the Hive connection mode to access Hive as a source or target.
3. Use the Hive connection to create a data object to write to Hive.
4. Add the data object to a mapping and configure the mapping to run in the Hive environment.

You can define the following types of objects in a Hive mapping:

- A read data object to read data from Hive
- Transformations
- A target or an SQL data service. You can write to Hive if you run the mapping in a Hadoop cluster.

Validate and run the mapping. You can deploy the mapping and run it or add the mapping to a Mapping task in a workflow.

Hive Mapping Example

Your organization, HypoMarket Corporation, needs to analyze customer data. Create a mapping that reads all the customer records. Create an SQL data service to make a virtual database available for end users to query.

You can use the following objects in a Hive mapping:

**Hive input**

The input file is a Hive table that contains the customer names and contact details.

Create a relational data object. Configure the Hive connection and specify the table that contains the customer data as a resource for the data object. Drag the data object into a mapping as a read data object.

**SQL Data Service output**

Create an SQL data service in the Developer tool. To make it available to end users, include it in an application, and deploy the application to a Data Integration Service. When the application is running, connect to the SQL data service from a third-party client tool by supplying a connect string.

You can run SQL queries through the client tool to access the customer data.

Social Media Mappings

Create mappings to read social media data from sources such as Facebook and LinkedIn.

You can extract social media data and load them to a target in the native environment only. You can choose to parse this data or use the data for data mining and analysis.

To process or analyze the data in Hadoop, you must first move the data to a relational or flat file target and then run the mapping in the Hadoop cluster.

You can use the following Informatica adapters in the Developer tool:

- PowerExchange for DataSift
- PowerExchange for Facebook
- PowerExchange for LinkedIn
Twitter Mapping Example

Your organization, Hypomarket Corporation, needs to review all the tweets that mention your product "HypoBasket" with a positive attitude since the time you released the product in February 2012.

Create a mapping that identifies tweets that contain the word HypoBasket and writes those records to a table.

You can use the following objects in a Twitter mapping:

**Twitter input**
- The mapping source is a Twitter data object that contains the resource Search.
- Create a physical data object and add the data object to the mapping. Add the Search resource to the physical data object. Modify the query parameter with the following query:
  
  `QUERY=HypoBasket:)&since:2012-02-01`

**Sorter transformation**
- Optionally, sort the data based on the timestamp.
- Add a Sorter transformation to the mapping. Specify the timestamp as the sort key with direction as ascending.

**Mapping output**
- Add a relational data object to the mapping as a target.

After you run the mapping, Data Integration Service writes the extracted tweets to the target table. You can use text analytics and sentiment analysis tools to analyze the tweets.
This chapter includes the following topics:

- Mappings in a Hive Environment Overview, 19
- Sources in a Hive Environment, 20
- Targets in a Hive Environment, 21
- Transformations in a Hive Environment, 23
- Functions in a Hive Environment, 25
- Mappings in a Hive Environment, 26
- Datatypes in a Hive Environment, 26
- Workflows in a Hive Environment, 27
- Configuring a Mapping to Run in a Hive Environment, 27
- Hive Execution Plan, 28
- Monitoring a Mapping, 28
- Logs, 29
- Troubleshooting a Mapping in a Hive Environment, 29

Mappings in a Hive Environment Overview

You can run a mapping on a Hadoop cluster. The Data Integration Service can push mappings that are imported from PowerCenter or developed in the Developer tool to a Hadoop cluster. You can run standalone mappings or mappings that are a part of a workflow.

When you run a mapping on a Hadoop cluster, you must configure a Hive validation environment, a Hive runtime environment, and a Hive connection for the mapping. Validate the mapping to ensure you can push the mapping logic to Hadoop. After you validate a mapping for the Hive environment, you can run the mapping.

To run a mapping on a Hadoop cluster, complete the following steps:

1. In the Developer tool, create a Hive connection.
2. Create a mapping in the Developer tool or import a mapping from PowerCenter.
3. Configure the mapping to run in a Hive environment.
4. Validate the mapping.
5. Optionally, include the mapping in a workflow.
6. Run the mapping or workflow.
When you run the mapping, the Data Integration Service converts the mapping to a Hive execution plan that runs on a Hadoop cluster. You can view the Hive execution plan using the Developer tool or the Administrator tool.

The Data Integration Service has a Hive executor that can process the mapping. The Hive executor simplifies the mapping to an equivalent mapping with a reduced set of instructions and generates a Hive execution plan. The Hive execution plan is a series of Hive queries. The Hive execution plan contains tasks to start the mapping, run the mapping, and clean up the temporary tables and files. You can view the Hive execution plan that the Data Integration Service generates before you run the mapping.

You can monitor Hive queries and the Hadoop jobs associated with a query in the Administrator tool. The Data Integration Service logs messages from the DTM, Hive session, and Hive tasks in the runtime log files.

Sources in a Hive Environment

Due to the differences between the native environment and a Hive environment, you can only push certain sources to a Hive environment. Some of the sources that are valid in mappings in a Hive environment have restrictions.

You can run mappings with the following sources in a Hive environment:

- IBM DB2
- Flat file
- HDFS complex file
- HDFS flat file
- Hive
- ODBC
- Oracle

Flat File Sources

Flat file sources are valid in mappings in a Hive environment with some restrictions. A mapping with a flat file source can fail to run in certain cases.

Flat file sources are valid in mappings in a Hive environment with the following restrictions:

- You cannot use a command to generate or transform flat file data and send the output to the flat file reader at runtime.
- You cannot use an indirect source type.
- The row size in a flat file source cannot exceed 190 MB.
Hive Sources

Hive sources are valid in mappings in a Hive environment with some restrictions.

Hive sources are valid in mappings in a Hive environment with the following restrictions:

- The Data Integration Service can run pre-mapping SQL commands against the source database before it reads from a Hive source. When you run a mapping with a Hive source in a Hive environment, references to local path in pre-mapping SQL commands are relative to the Data Integration Service node. When you run a mapping with a Hive source in the native environment, references to local path in pre-mapping SQL commands are relative to the Hive server node.
- A mapping fails to validate when you configure post-mapping SQL commands. The Data Integration Service does not run post-mapping SQL commands against a Hive source.
- A mapping fails to run when you have Unicode characters in a Hive source definition.

Relational Sources

Relational sources are valid in mappings in a Hive environment with certain restrictions.

The Data Integration Service does not run pre-mapping SQL commands or post-mapping SQL commands against relational sources. You cannot validate and run a mapping with PreSQL or PostSQL properties for a relational source in a Hive environment.

The Data Integration Service can use multiple partitions to read from the following relational sources:

- IBM DB2
- Oracle

Note: You do not have to set maximum parallelism for the Data Integration Service to use multiple partitions in the Hive environment.

Targets in a Hive Environment

Due to the differences between the native environment and a Hive environment, you can push only certain targets to a Hive environment. Some of the targets that are valid in mappings in a Hive environment have restrictions.

You can run mappings with the following targets in a Hive environment:

- IBM DB2
- Flat file
- HDFS flat file
- Hive
- ODBC
- Oracle
- Teradata
Flat File Targets

Flat file targets are valid in mappings in a Hive environment with some restrictions.

Flat file targets are valid in mappings in a Hive environment with the following restrictions:

- The Data Integration Service truncates the target files and reject files before writing the data. When you use a flat file target, you cannot append output data to target files and reject files.
- The Data Integration Service can write to a file output for a flat file target. When you have a flat file target in a mapping, you cannot write data to a command.

HDFS Flat File Target

HDFS flat file targets are valid in mappings in a Hive environment with some restrictions.

When you use a HDFS flat file target in a mapping, you must specify the full path that includes the output file directory and file name. The Data Integration Service may generate multiple output files in the output directory when you run the mapping in a Hive environment.

Hive Targets

Hive targets are valid in mappings in a Hive environment with some restrictions.

Hive targets are valid in mappings in a Hive environment with the following restrictions:

- The Data Integration Service does not run pre-mapping or post-mapping SQL commands against the target database for a Hive target. You cannot validate and run a mapping with PreSQL or PostSQL properties for a Hive target.
- A mapping fails to run if the Hive target definition differs in the number and order of the columns from the relational table in the Hive database.
- You must choose to truncate the target table to overwrite data to a Hive table with Hive version 0.7. The Data Integration Service ignores write, update override, delete, insert, and update strategy properties when it writes data to a Hive target.
- A mapping fails to run when you use Unicode characters in a Hive target definition.

Relational Targets

Relational targets are valid in mappings in a Hive environment with certain restrictions.

The Data Integration Service does not run pre-mapping SQL commands or post-mapping SQL commands against relational targets in a Hive environment. You cannot validate and run a mapping with PreSQL or PostSQL properties for a relational target in a Hive environment.

The Data Integration Service can use multiple partitions to write to Oracle relational targets.

**Note:** You do not have to set maximum parallelism for the Data Integration Service to use multiple partitions in the Hive environment.
Transformations in a Hive Environment

Due to the differences between native and Hive environment only certain transformations are valid or valid with restrictions in the Hive environment. The Data Integration Service does not process transformations that contain functions, expressions, datatypes, and variable fields that are not valid in a Hive environment.

Use the following rules and guidelines for transformations:

**Aggregator**

An Aggregator transformation with pass-through fields is valid if they are group-by fields.

**Case Convertor**

The Data Integration Service can push a Case Converter transformation to Hadoop.

**Data Masking**

You cannot use the following data masking techniques in mapping logic run on Hadoop clusters:

- Repeatable expression masking
- Unique repeatable substitution masking

**Data Processor**

The following limitations apply when a Data Processor transformation directly connects to a complex file reader:

- Ports cannot be defined as file.
- Input port must be defined as binary.
- Output port cannot be defined as binary.
- A Streamer must be defined as startup component.
- Pass-through ports cannot be used.
- Additional input ports cannot be used.

**Expression**

An Expression transformation with a user-defined function returns a null value for rows that have an exception error in the function.

The Data Integration Service returns an infinite or a NaN (not a number) value when you push transformation logic to Hadoop for expressions that result in numerical errors. For example:

- Divide by zero
- SQRT (negative number)
- ASIN (out-of-bounds number)

In the native environment, the expressions that result in numerical errors return null values and the rows do not appear in the output.

**Filter**

The Data Integration Service can push a Filter transformation to Hadoop.

**Java**

You must copy external JAR files that a Java transformation requires to the Informatica installation directory in the Hadoop cluster nodes at the following location: \$HADOOP\$NODE\$INFA\$HOME/services/shared/jars/platform/dtm/

You can optimize the transformation for faster processing when you enable an input port as a partition key and sort key. The data is partitioned across the reducer tasks and the output is partially sorted.
The following limitations apply to the Transformation Scope property:

- The value Transaction for transformation scope is not valid.
- If transformation scope is set to Row, a Java transformation is run by mapper script.
- If you enable an input port for partition Key, the transformation scope is set to All Input. When the transformation scope is set to All Input, a Java transformation is run by the reducer script and you must set at least one input field as a group-by field for the reducer key.

You can enable the Stateless advanced property when you run mappings in a Hive environment.

The Java code in the transformation cannot write output to standard output when you push transformation logic to Hadoop. The Java code can write output to standard error which appears in the log files.

Joiner

A Joiner transformation cannot contain inequality joins in the outer join condition.

Lookup

The following limitations apply to Lookup transformations:

- An unconnected Lookup transformation is not valid.
- You cannot configure an uncached lookup source.
- You cannot configure a persistent lookup cache for the lookup source.
- You cannot use a Hive source for a relational lookup source.
- When you run mappings that contain Lookup transformations, the Data Integration Service creates lookup cache Jar files. Hive copies the lookup cache JAR files to the following temporary directory: `/tmp/<user_name>/hive_resources`. The Hive parameter `hive.downloaded.resources.dir` determines the location of the temporary directory. You can delete the lookup cache JAR files specified in the LDTM log after the mapping completes to retrieve disk space.

Merge

The Data Integration Service can push a Merge transformation to Hadoop.

Rank

A comparison is valid if it is case sensitive.

Router

The Data Integration Service can push a Router transformation to Hadoop.

Sorter

The Data Integration Service ignores the Sorter transformation when you push mapping logic to Hadoop.

SQL

The Data Integration Service can push SQL transformation logic to Hadoop.

You cannot use a Hive connection.

Union

The custom source code in the transformation cannot write output to standard output when you push transformation logic to Hadoop. The custom source code can write output to standard error, that appears in the runtime log files.
Functions in a Hive Environment

Some transformation language functions that are valid in the native environment are not valid or have limitations in a Hive environment.

The following table describes the functions that are not valid or have limitations in a Hive environment:

<table>
<thead>
<tr>
<th>Name</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>String argument is not valid.</td>
</tr>
<tr>
<td>AES_DECRYPT</td>
<td>Not valid</td>
</tr>
<tr>
<td>AES_ENCRYPT</td>
<td>Not valid</td>
</tr>
<tr>
<td>COMPRESS</td>
<td>Not valid</td>
</tr>
<tr>
<td>CRC32</td>
<td>Not valid</td>
</tr>
<tr>
<td>CUME</td>
<td>Not valid</td>
</tr>
<tr>
<td>DEC_BASE64</td>
<td>Not valid</td>
</tr>
<tr>
<td>DECOMPRESS</td>
<td>Not valid</td>
</tr>
<tr>
<td>ENC_BASE64</td>
<td>Valid if the argument is UUID4().</td>
</tr>
<tr>
<td>ERROR</td>
<td>String argument is not valid.</td>
</tr>
<tr>
<td>FIRST</td>
<td>Not valid</td>
</tr>
<tr>
<td>LAST</td>
<td>Not valid</td>
</tr>
<tr>
<td>MAX (Dates)</td>
<td>Not valid</td>
</tr>
<tr>
<td>MD5</td>
<td>Not valid</td>
</tr>
<tr>
<td>MIN (Dates)</td>
<td>Not valid</td>
</tr>
<tr>
<td>MOVINGAVG</td>
<td>Not valid</td>
</tr>
<tr>
<td>MOVINGSUM</td>
<td>Not valid</td>
</tr>
<tr>
<td>UUID4()</td>
<td>Valid as an argument in UUID_UNPARSE or ENC_BASE64.</td>
</tr>
<tr>
<td>UUID_UNPARSE(Binary)</td>
<td>Valid if the argument is UUID4().</td>
</tr>
</tbody>
</table>
Mappings in a Hive Environment

You can run mappings in a Hive environment. Some differences in processing and configuration apply when you run mappings in a Hive environment.

The following processing differences apply to mappings in a Hive environment:

• A mapping is run in low precision mode. The Data Integration Service ignores high precision mode in a Hive environment. Mappings that require high precision mode may fail to run in a Hive environment.

• In a Hive environment, sources that have data errors in a column result in a null value for the column. In the native environment, the Data Integration Service does not process the rows that have data errors in a column.

• When you cancel a mapping that reads from a flat file source, the file copy process that copies flat file data to HDFS may continue to run. The Data Integration Service logs the command to kill this process in the Hive session log, and cleans up any data copied to HDFS. Optionally, you can run the command to kill the file copy process.

The following configuration differences apply to mappings in a Hive environment:

• Set the optimizer level to none or minimal if a mapping validates but fails to run. If you set the optimizer level to use cost-based or semi-join optimization methods, the Data Integration Service ignores this at run-time and uses the default.

• Mappings that contain a Hive source or a Hive target must use the same Hive connection to push the mapping to Hadoop.

• The Data Integration Service ignores the data file block size configured for HDFS files in the hdfs-site.xml file. The Data Integration Service uses a default data file block size of 64 MB for HDFS files. To change the data file block size, copy /usr/lib/hadoop/conf/hdfs-site.xml to the following location in the Hadoop distribution directory for the Data Integration Service node: /opt/Informatica/services/shared/hadoop/[Hadoop_distribution_name]/conf. You can also update the data file block size in the following file: /opt/Informatica/services/shared/hadoop/[Hadoop_distribution_name]/conf/hive-default.xml.

Datatypes in a Hive Environment

Due to the differences between the native environment and a Hive environment, some variations apply in the processing and validity of datatypes when you push datatypes to a Hive environment.

The following variations apply in datatype processing and validity:

• A Binary datatype in a field or an expression function is not valid. If a transformation has a port with a Binary datatype that is not used in the mapping, you can validate and run the mapping in a Hive environment.

• A high precision Decimal datatype is not valid. A mapping is run in low precision mode in a Hive environment.

• The results of arithmetic operations on floating point types, such as a Double or a Decimal, can vary up to 0.1 percent between the native environment and a Hive environment.

• Hive complex datatypes in a Hive source or Hive target are not valid.

• When the Data Integration Service converts a decimal with a precision of 10 and a scale of 3 to a string datatype and writes to a flat file target, the results can differ between the native environment and a Hive environment. For example, in a Hive environment, HDFS writes the output string for the decimal 19711025 as '19711025.000'.
with a precision of 10 and a scale of 3 as 1971. In the native environment, the flat file writer sends the output string for the decimal 19711025 with a precision of 10 and a scale of 3 as 1971.000.

- Hive uses a maximum or minimum value for BigInt and Integer datatypes when there is data overflow during datatype conversion. Mapping results can vary between the native and Hive environment when there is data overflow during datatype conversion for BigInt and Integer datatypes.

### Variable Ports in a Hive Environment

A transformation that contains a stateful variable port is not valid in a Hive environment.

A stateful variable port refers to values from previous rows.

### Workflows in a Hive Environment

You can add a mapping configured to run in a Hive environment to a Mapping task in a workflow. When you deploy and run the workflow, the Mapping task runs the mapping.

You might want to run a mapping from a workflow so that you can run multiple mappings sequentially, make a decision during the workflow, or send an email notifying users of the workflow status. Or, you can develop a workflow that runs commands to perform steps before and after the mapping runs.

When a Mapping task runs a mapping configured to run in a Hive environment, do not assign the Mapping task outputs to workflow variables. Mappings that run in a Hive environment do not provide the total number of target, source, and error rows. When a Mapping task includes a mapping that runs in a Hive environment, the task outputs contain a value of zero (0).

### Configuring a Mapping to Run in a Hive Environment

You can use the Developer tool to configure a mapping to run in a Hive environment. To configure a mapping, you must specify a Hive validation environment, a Hive run-time environment, and a Hive connection.

Configure the following pre-requisites in the file `<Informatica Client Installation Directory>\clients\DeveloperClient\DeveloperCore.ini`:

- When you use Hortonworks 1.3.2 distribution, you must modify the variable INFA_HADOOP_DIST_DIR to hadoop\hortonworks_1.3.2.

1. Open the mapping in the Developer tool.
2. In the Properties view, select the Advanced tab.
3. Select Hive as the value for the validation environment.
4. Select a Hive version.
5. In the Run-time tab, select Hive-<version> as the execution environment.
6. Select a Hive connection.
Hive Execution Plan

The Data Integration Service generates a Hive execution plan for a mapping when you run a mapping in a Hive environment. A Hive execution plan is a series of Hive tasks that the Hive executor generates after it processes a mapping for a Hive environment.

Hive Execution Plan Details

You can view the details of a Hive execution plan for a mapping from the Developer tool.

The following table describes the properties of a Hive execution plan:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Script Name</td>
<td>Name of the Hive script.</td>
</tr>
<tr>
<td>Script</td>
<td>Hive script that the Data Integration Service generates based on the mapping logic.</td>
</tr>
<tr>
<td>Depends On</td>
<td>Tasks that the script depends on. Tasks include other scripts and Data Integration Service tasks, like the Start task.</td>
</tr>
</tbody>
</table>

Viewing the Hive Execution Plan for a Mapping

You can view the Hive execution plan for a mapping that runs in a Hive environment. You do not have to run the mapping to view the Hive execution plan in the Developer tool.

**Note:** You can also view the Hive execution plan in the Administrator tool.

1. In the Developer tool, open the mapping.
2. Select the **Data Viewer** tab.
3. Select **Show Execution Plan**.
   - The **Data Viewer** tab shows the the details for the Hive execution plan.

Monitoring a Mapping

You can monitor a mapping that is running on a Hadoop cluster.

1. Open the **Monitoring** tab in the Administrator tool.
2. Select **Jobs** in the Navigator.
3. Select the mapping job.
4. Click the **View Logs for Selected Object** button to view the run-time logs for the mapping.
   - The log shows the results of the Hive queries run by the Data Integration Service. This includes the location of Hive session logs and Hive session history file.
5. To view the Hive execution plan for the mapping, select the **Hive Query Plan** view.
6. To view each script and query included in the Hive execution plan, expand the mapping job node, and select the Hive script or query.
Logs

The Data Integration Service generates log events when you run a mapping in a Hive environment.

You can view log events relating to different types of errors such as Hive connection failures, Hive query failures, Hive command failures, or other Hadoop job failures. You can find the information about these log events in the following log files:

LDTM log

The Logical DTM logs the results of the Hive queries run for the mapping. You can view the Logical DTM log from the Developer tool or the Administrator tool for a mapping job.

Hive session log

For every Hive script in the Hive execution plan for a mapping, the Data Integration Service opens a Hive session to run the Hive queries. A Hive session updates a log file in the following directory on the Data Integration Service node: `<InformaticaInstallationDir>/tomcat/bin/disTemp/`. The full path to the Hive session log appears in the LDTM log.

Troubleshooting a Mapping in a Hive Environment

When I run a mapping with a Hive source or a Hive target on a different cluster, the Data Integration Service fails to push the mapping to Hadoop with the following error: Failed to execute query [exec0_query_6] with error code [10], error message [FAILED: Error in semantic analysis: Line 1:181 Table not found customer_eur], and SQL state [42000].

When you run a mapping in a Hive environment, the Hive connection selected for the Hive source or Hive target, and the mapping must be on the same Hive metastore.
Profiles Overview

You can run a profile on HDFS and Hive data sources in the Hadoop environment. The Hadoop environment helps improve the performance. The run-time environment, native Data Integration Service or Hadoop, does not affect the profile results.

You can run a column profile, rule profile, and data domain discovery on a single data object profile in the Hadoop environment. You can perform these profiling capabilities on both native and Hadoop data sources. A native data source is a non-Hadoop source, such as a flat file, relational source, or mainframe source. A Hadoop data source can be either a Hive or HDFS source.

If you use Informatica Developer, you can choose either native or Hadoop run-time environment to run a profile. If you choose the Hadoop environment, the Developer tool sets the run-time environment in the profile definition. Informatica Analyst supports native environment that uses the Data Integration Service.

You run a profile in the Hadoop run-time environment from the Developer tool. You validate a data source to run the profile in both native and Hadoop environments. To validate the profile run in the Hadoop environment, you must select a Hive connection. You can then choose to run the profile in either native or Hadoop run-time environment.

You can view the Hive query plan in the Administrator tool. The Hive query plan consists of one or more scripts that the Data Integration Service generates based on the logic defined in the profile. Each script contains Hive queries that run against the Hive database. One query contains details about the MapReduce job. The remaining queries perform other actions such as creating and dropping tables in the Hive database.

You can use the Monitoring tab of the Administrator tool to monitor a profile and Hive statements running on Hadoop. You can expand a profile job to view the Hive queries generated for the profile. You can also view
the run-time log for each profile. The log shows run-time details, such as the time each task runs, the Hive queries that run on Hadoop, and errors that occur.

The Monitoring tab contains the following views:

Properties view

The Properties view shows properties about the selected profile.

Hive Query Plan view

The Hive Query Plan view shows the Hive query plan for the selected profile.

Native and Hadoop Environments

When you run a profile in the native environment, the Analyst tool or Developer tool submits the profile jobs to the Profiling Service Module. The Profiling Service Module then breaks down the profile jobs into a set of mappings. The Data Integration Service runs these mappings and writes the profile results to the profile warehouse.

The native environment runs the mappings on the same machine where the Data Integration Service runs. The Hadoop environment runs the mappings on a Hadoop cluster. The Data Integration Service pushes the mapping execution to the Hadoop cluster through a Hive connection. This environment makes all the sources, transformations, and Hive and HDFS sources available for profile run.

If you choose a native source for the Hadoop run-time environment, the Data Integration Service runs the profile on Hadoop. You cannot run a Hadoop data source in the native run-time environment.

Supported Data Source and Run-time Environments

In the Developer tool, you can run a profile on native, Hive, and HDFS data sources. You can run a profile on both Hive and HDFS sources in the Hadoop environment.

The following table describes the combination of data source types and run-time environments that Data Explorer supports:

<table>
<thead>
<tr>
<th>Data Source Type</th>
<th>Run-time Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native sources such as flat files, relational sources, and mainframes</td>
<td>Native, Hadoop</td>
</tr>
<tr>
<td>Hive</td>
<td>Hadoop</td>
</tr>
<tr>
<td>HDFS</td>
<td>Hadoop</td>
</tr>
</tbody>
</table>

You cannot run some of the profile definitions in either the native or Hadoop environment.
The following table describes some of the run-time scenarios and whether you can run the profile in different run-time environments:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Hadoop Run-time Environment</th>
<th>Native Run-time Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running a profile on a Hive or HDFS source within a mapping specification.</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Running a profile on a mapping specification with a Hive or HDFS data source.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Running a profile on a logical data object with a Hive or HDFS data source.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Running a column profile on a mapping or mapplet object with a Hive or Hadoop source.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Comparing the column profile results of two objects in a mapping or mapplet object with a Hive or HDFS source.</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Run-time Environment Setup and Validation

By default, all profiles run in the native run-time environment. You can change the run-time environment to Hadoop in the Developer tool and run a profile. Before you run a profile, you need to verify whether the validation settings in the profile definition match its run-time requirements.

The validation settings determine whether the profile definition suits the native run-time environment, Hadoop run-time environment, or both. The steps to complete the run-time environment setup and validation are as follows:

1. Choose the validation environments. Validation environments are the environments that you want to set up for the profile run. The Developer tool validates the data sources and transformations for these environments. You must choose at least one of the environments. If you choose both environments, you must choose the run-time environment for the profile.

2. Choose the run-time environment. When you choose the run-time environment, the Developer tool saves one of the associated validation environments for profile run. If you choose Hadoop, you must select a Hive connection. The Hive connection helps the Data Integration Service communicate with the Hadoop cluster to push down the mapping execution from the Data Integration Service to the Hadoop cluster.

The validation environments determine whether the sources and transformations that any of the source rules and data domains may contain are valid for the environments. The Developer tool validates a profile definition before you run it.
The following table describes the validation environment settings that you can configure for a profile:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native (Data Integration Service)</td>
<td>The Data Integration Service runs the profile.</td>
</tr>
<tr>
<td>Hadoop</td>
<td>Runs the profile in the Hadoop environment. If you select this option, you must specify the Hive connection.</td>
</tr>
<tr>
<td>Hive connection</td>
<td>The Hive connection to run a profile in the Hadoop environment.</td>
</tr>
</tbody>
</table>

You can specify both native and Hadoop options when you set up the validation environments for a profile. You choose either **Native** or **Hadoop** as the run-time environment.

### Run-time Environment and Profile Performance

In general, you run a profile on Hadoop data in the Hadoop run-time environment. For non-Hadoop data, profiles on smaller data sources run faster in the native run-time environment.

You can run a profile on bigger data sources in the Hadoop run-time environment. In addition to the data size, you also need to consider many other factors such as the network configuration, Data Integration Service configuration, and Hadoop cluster configuration. Unless you need to run non-Hadoop data in the Hadoop run-time environment at a later stage, you run a profile on data in the environment it resides.

### Profile Types on Hadoop

You can run a column profile, data domain profile, and column profile with rules in the Hadoop environment.

You can run a column profile in the Hadoop environment to determine the characteristics of source columns such as value frequency, percentages, patterns, and datatypes. Run a data domain profile in the Hadoop environment to discover source column data that match predefined data domains based on data and column name rules. You can also run a profile that has associated rules in the Hadoop environment.

**Note:** Random sampling may not apply when you run a column profile in the Hadoop environment.

### Column Profiles on Hadoop

You can import a native or Hadoop data source into the Developer tool and then run a column profile on it. When you create a column profile, you select the columns, set up filters, and sampling options. Column profile results include value frequency distribution, unique values, null values, and datatypes.

Complete the following steps to run a column profile on Hadoop.

1. Open a connection in the Developer tool to import the native or Hadoop source.
2. Import the data source as a data object. The Developer tool saves the data object in the Model repository.
3. Create a profile on the imported data object.
4. Set up the configuration options. These options include validation environment settings, run-time settings, and the Hive connection.
5. Run the profile to view the results.

Rule Profiles on Hadoop

You can run profiles on Hadoop that apply business rules to identify problems in the source data. In the Developer tool, you can create a mapplet and validate the mapplet as a rule for reuse. You can also add a rule to a column profile on Hadoop.

You cannot run profiles that contain stateful functions, such as MOVINGAVG, MOVINGSUM, or COMPRESS.

For more information about stateful functions, see the Mappings in a Hive Environment chapter.

Data Domain Discovery on Hadoop

Data domain discovery is the process of discovering logical datatypes in the data sources based on the semantics of data. You can run a data domain profile on Hadoop and view the results in the Developer tool.

Data domain discovery results display statistics about columns that match data domains, including the percentage of matching column data and whether column names match data domains. You can drill down the results further for analysis, verify the results on all the rows of the data source, and add the results to a data model from the profile model.

Running a Single Data Object Profile on Hadoop

After you set up the validation and run-time environments for a profile, you can run the profile to view its results.

1. In the Object Explorer view, select the data object you want to run a profile on.
2. Click File > New > Profile.
   The profile wizard appears.
3. Select Profile and click Next.
4. Enter a name and description for the profile and verify the project location. If required, browse to a new location.
   Verify that Run Profile on finish is selected.
5. Click Next.
6. Configure the column profiling and domain discovery options.
7. Click Run Settings.
   The Run Settings pane appears.
8. Select Hive as the validation environment.
   You can select both Native and Hive as the validation environments.
9. Select Hive as the run-time environment.
10. Select a Hive connection.
11. Click Finish.
Running Multiple Data Object Profiles on Hadoop

You can run a column profile on multiple data source objects. The Developer tool uses default column profiling options to generate the results for multiple data sources.

1. In the Object Explorer view, select the data objects you want to run a profile on.
2. Click File > New > Profile to open the New Profile wizard.
3. Select Multiple Profiles and click Next.
4. Select the location where you want to create the profiles. You can create each profile at the same location of the data object, or you can specify a common location for the profiles.
5. Verify that the names of the data objects you selected appear within the Data Objects section.
   Optionally, click Add to add another data object.
6. Optionally, specify the number of rows to profile, and choose whether to run the profile when the wizard completes.
7. Click Next.
   The Run Settings pane appears. You can specify the Hive settings.
8. Select Hive and select a Hive connection.
   You can select both Native and Hive as the validation environments.
9. In the Run-time Environment field, select Hive.
10. Click Finish.
11. Optionally, enter prefix and suffix strings to add to the profile names.
12. Click OK.

Monitoring a Profile

You can monitor a profile that is running on Hadoop.

1. Open the Monitoring tab in the Administrator tool.
2. Select Jobs in the Navigator.
3. Select the profiling job.
4. Click the View Logs for Selected Object button to view the run-time logs for the profile.
   The log shows all the hive queries that the Data Integration Service ran on the Hadoop cluster.
5. To view the Hive query plan for the profile, select the Hive Query Plan view.
   You can also view the Hive query plan in the Developer tool.
6. To view each script and query included in the Hive query plan, expand the profiling job node, and select the Hive script or query.
Viewing Profile Results

You can view the column profile and data domain discovery results after you run a profile on Hadoop.

1. In the Object Explorer view, select the profile you want to view the results for.
2. Right-click the profile and select Run Profile.
   The Run Profile dialog box appears.
3. Click the Results tab, if not selected already, in the right pane.
   You can view the column profile and data domain discovery results in separate panes.

Troubleshooting

Can I drill down on profile results if I run a profile in the Hadoop environment?
Yes, except for profiles in which you have set the option to drill down on staged data.

I get the following error message when I run a profile in the Hadoop environment: "[LDTM_1055] The Integration Service failed to generate a Hive workflow for mapping [Profile_CUSTOMER_INFO12_14258652520457390]." How do I resolve this?
This error can result from a data source, rule transformation, or run-time environment that is not supported in the Hadoop environment. For more information about objects that are not valid in the Hadoop environment, see the Mappings in a Hive Environment chapter.

You can change the data source, rule, or run-time environment and run the profile again. View the profile log file for more information on the error.

I see "N/A" in the profile results for all columns after I run a profile. How do I resolve this?
Verify that the profiling results are in the profiling warehouse. If you do not see the profile results, verify that the database path is accurate in the HadoopEnv.properties file. You can also verify the database path from the Hadoop job tracker on the Monitoring tab of the Administrator tool.

After I run a profile on a Hive source, I do not see the results. When I verify the Hadoop job tracker, I see the following error when I open the profile job: "XML Parsing Error: no element found." What does this mean?
The Hive data source does not have any record and is empty. The data source must have a minimum of one row of data for successful profile run.

After I run a profile on a Hive source, I cannot view some of the column patterns. Why?
When you import a Hive source, the Developer tool sets the precision for string columns to 4000. The Developer tool cannot derive the pattern for a string column with a precision greater than 255. To resolve this issue, set the precision of these string columns in the data source to 255 and run the profile again.

When I run a profile on large Hadoop sources, the profile job fails and I get an "execution failed" error. What can be the possible cause?
One of the causes can be a connection issue. Perform the following steps to identify and resolve the connection issue:
1. Open the Hadoop job tracker.
2. Identify the profile job and open it to view the MapReduce jobs.
3. Click the hyperlink for the failed job to view the error message. If the error message contains the text "java.net.ConnectException: Connection refused", the problem occurred because of an issue with the Hadoop cluster. Contact your network administrator to resolve the issue.
Native Environment Optimization Overview

You can optimize the native environment to increase performance. You can also enable high availability to ensure that the domain can continue running despite temporary network, hardware, or service failures.

You can also run a Model repository mapping with partitioning to increase performance. When you run a partitioned mapping, the Data Integration Service performs the extract, transformation, and load for each partition in parallel.

You can configure high availability for the domain. High availability eliminates a single point of failure in a domain and provides minimal service interruption in the event of failure.

Processing Big Data on Partitions

You can run a Model repository mapping with partitioning to increase performance. When you run a mapping configured with partitioning, the Data Integration Service performs the extract, transformation, and load for each partition in parallel.

Mappings that process large data sets can take a long time to process and can cause low data throughput. When you configure partitioning, the Data Integration Service uses additional threads to process the mapping which can increase performance.

Partitioned Model Repository Mappings

You can enable the Data Integration Service process to use multiple partitions to process Model repository mappings.

If the Data Integration Service process runs on a node with multiple CPUs, you can enable the Data Integration Service process to maximize parallelism when it runs mappings. When you maximize parallelism,
the Data Integration Service dynamically divides the underlying data into partitions and processes all of the partitions concurrently.

Optionally, you can disable partitioning for the mapping or decrease the number of maximum parallel threads for the mapping. When maximum parallelism is set to different values for the Data Integration Service process and the mapping, the Data Integration Service uses the minimum value.

For more information, see the Informatica Application Service Guide and the Informatica Mapping Guide.

Partition Optimization

You can optimize the partitioning of Model repository mappings to increase performance. You can add more partitions, select the best performing partition types, use more CPUs, and optimize the source or target database for partitioning.

To optimize partitioning, perform the following tasks:

Increase the number of partitions.

When you configure Model repository mappings, you increase the number of partitions when you increase the maximum parallelism value for the Data Integration Service process or the mapping.

Increase the number of partitions to enable the Data Integration Service to create multiple connections to sources and process partitions of source data concurrently. Increasing the number of partitions increases the number of threads, which also increases the load on the Data Integration Service nodes. If the Data Integration Service node or nodes contain ample CPU bandwidth, processing rows of data concurrently can increase performance.

Note: If you use a single-node Integration Service and the Integration Service uses a large number of partitions in a session or mapping that processes large amounts of data, you can overload the system.

Use multiple CPUs.

If you have a symmetric multi-processing (SMP) platform, you can use multiple CPUs to concurrently process partitions of data.

Optimize the source database for partitioning.

You can optimize the source database for partitioning. For example, you can tune the database, enable parallel queries, separate data into different tablespaces, and group sorted data.

Optimize the target database for partitioning.

You can optimize the target database for partitioning. For example, you can enable parallel inserts into the database, separate data into different tablespaces, and increase the maximum number of sessions allowed to the database.

High Availability

High availability eliminates a single point of failure in an Informatica domain and provides minimal service interruption in the event of failure. When you configure high availability for a domain, the domain can continue running despite temporary network, hardware, or service failures. You can configure high availability for the domain, application services, and application clients.

The following high availability components make services highly available in an Informatica domain:

- Resilience. An Informatica domain can tolerate temporary connection failures until either the resilience timeout expires or the failure is fixed.
• Restart and failover. A process can restart on the same node or on a backup node after the process becomes unavailable.

• Recovery. Operations can complete after a service is interrupted. After a service process restarts or fails over, it restores the service state and recovers operations.

When you plan a highly available Informatica environment, configure high availability for both the internal Informatica components and systems that are external to Informatica. Internal components include the domain, application services, application clients, and command line programs. External systems include the network, hardware, database management systems, FTP servers, message queues, and shared storage.

High availability features for the Informatica environment are available based on your license.
Datatype Reference

This appendix includes the following topics:

- Datatype Reference Overview, 41
- Hive Complex Datatypes, 41
- Hive Data Types and Transformation Data Types, 42

Datatype Reference Overview

Informatica Developer uses the following datatypes in Hive mappings:

- Hive native datatypes. Hive datatypes appear in the physical data object column properties.
- Transformation datatypes. Set of datatypes that appear in the transformations. They are internal datatypes based on ANSI SQL-92 generic datatypes, which the Data Integration Service uses to move data across platforms. Transformation datatypes appear in all transformations in a mapping.

When the Data Integration Service reads source data, it converts the native datatypes to the comparable transformation datatypes before transforming the data. When the Data Integration Service writes to a target, it converts the transformation datatypes to to the comparable native datatypes.

Hive Complex Datatypes

Hive complex datatypes such as arrays, maps, and structs are a composite of primitive or complex datatypes. Informatica Developer represents the complex datatypes with the string dataype and uses delimiters to separate the elements of the complex datatype.

Note: Hive complex datatypes in a Hive source or Hive target are not supported when you run mappings in a Hadoop cluster.
The following table describes the transformation types and delimiters that are used to represent the complex datatypes:

<table>
<thead>
<tr>
<th>Complex Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td>The elements in the array are of string datatype. Each element of the array is delimited by commas. For example, an array of fruits is represented as [apple, banana, orange].</td>
</tr>
<tr>
<td>Map</td>
<td>Maps contain key-value pairs and are represented as pairs of strings and integers delimited by the <code>=</code> character. Each pair of string and integer pair is delimited by commas. For example, a map of fruits is represented as [1=apple, 2=banana, 3=orange].</td>
</tr>
<tr>
<td>Struct</td>
<td>Struct are represented as pairs of strings and integers delimited by the <code>:</code> character. Each pair of string and integer pair is delimited by commas. For example, a map of fruits is represented as [1, apple].</td>
</tr>
</tbody>
</table>

**Hive Data Types and Transformation Data Types**

The following table lists the Hive data types that Data Integration Service supports and the corresponding transformation data types:

<table>
<thead>
<tr>
<th>Hive Data Type</th>
<th>Transformation Data Type</th>
<th>Range and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiny Int</td>
<td>Integer</td>
<td>-32,768 to 32,767</td>
</tr>
<tr>
<td>Integer</td>
<td>Integer</td>
<td>-2,147,483,648 to 2,147,483,647 Precision 10, scale 0</td>
</tr>
<tr>
<td>Bigint</td>
<td>Bigint</td>
<td>-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0</td>
</tr>
<tr>
<td>Decimal</td>
<td>Decimal</td>
<td>Precision 28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If a mapping is not enabled for high precision, the Data Integration Service converts all decimal values to double values.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If a mapping is enabled for high precision, the Data Integration Service converts decimal values with precision greater than 28 to double values.</td>
</tr>
<tr>
<td>Double</td>
<td>Double</td>
<td>Precision 15</td>
</tr>
<tr>
<td>Float</td>
<td>Double</td>
<td>Precision 15</td>
</tr>
<tr>
<td>String</td>
<td>String</td>
<td>1 to 104,857,600 characters</td>
</tr>
<tr>
<td>Hive Data Type</td>
<td>Transformation Data Type</td>
<td>Range and Description</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Boolean</td>
<td>Integer</td>
<td>1 or 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default transformation type for boolean is integer. You can also set this to string data type with values of True and False.</td>
</tr>
<tr>
<td>Arrays</td>
<td>String</td>
<td>1 to 104,857,600 characters</td>
</tr>
<tr>
<td>Struct</td>
<td>String</td>
<td>1 to 104,857,600 characters</td>
</tr>
<tr>
<td>Maps</td>
<td>String</td>
<td>1 to 104,857,600 characters</td>
</tr>
</tbody>
</table>
Glossary

**big data**
A set of data that is so large and complex that it cannot be processed through standard database management tools.

**Cloudera's Distribution Including Apache Hadoop (CDH)**
Cloudera's version of the open-source Hadoop software framework.

**CompressionCodec**
Hadoop compression interface. A codec is the implementation of a compression-decompression algorithm. In Hadoop, a codec is represented by an implementation of the CompressionCodec interface.

**DataNode**
An HDFS node that stores data in the Hadoop File System. An HDFS cluster can have more than one DataNode, with data replicated across them.

**Hadoop cluster**
A cluster of machines that is configured to run Hadoop applications and services. A typical Hadoop cluster includes a master node and several worker nodes. The master node runs the master daemons JobTracker and NameNode. A slave or worker node runs the DataNode and TaskTracker daemons. In small clusters, the master node may also run the slave daemons.

**Hadoop Distributed File System (HDFS)**
A distributed file storage system used by Hadoop applications.

**Hive environment**
An environment that you can configure to run a mapping or a profile on a Hadoop Cluster. You must configure Hive as the validation and run-time environment.

**Hive**
A data warehouse infrastructure built on top of Hadoop. Hive supports an SQL-like language called HiveQL for data summarization, query, and analysis.

**Hive executor**
A component of the DTM that can simplify and convert a mapping or a profile to a Hive execution plan that runs on a Hadoop cluster.
Hive execution plan
A series of Hive tasks that the Hive executor generates after it processes a mapping or a profile. A Hive execution plan can also be referred to as a Hive workflow.

Hive scripts
Script in Hive query language that contain Hive queries and Hive commands to run the mapping.

Hive task

JobTracker
A Hadoop service that coordinates map and reduce tasks and schedules them to run on TaskTrackers.

MapReduce
A programming model for processing large volumes of data in parallel.

MapReduce job
A unit of work that consists of the input data, the MapReduce program, and configuration information. Hadoop runs the MapReduce job by dividing it into map tasks and reduce tasks.

metastore
A database that Hive uses to store metadata of the Hive tables stored in HDFS. Metastores can be local, embedded, or remote.

NameNode
A node in the Hadoop cluster that manages the file system namespace, maintains the file system tree, and the metadata for all the files and directories in the tree.

native environment
The default environment in the Informatica domain that runs a mapping, a workflow, or a profile. The Integration Service performs data extraction, transformation, and loading.

run-time environment
The environment you configure to run a mapping or a profile. The run-time environment can be native or Hive.

stateful variable port
A variable port that refers to values from previous rows.

TaskTracker
A node in the Hadoop cluster that runs tasks such as map or reduce tasks. TaskTrackers send progress reports to the JobTracker.
validation environment

The environment you configure to validate a mapping or a profile. You validate a mapping or a profile to ensure that it can run in a run-time environment. The validation environment can be Hive, native, or both.
INDEX

A
architecture
  Hive environment processing 3

B
big data
  access 2
  big data processing
    example 4

C
column profiling on Hadoop
  overview 33
  connections
    HDFS 6
    Hive 6

D
data domain discovery on Hadoop
  overview 34
  data types
    Hive 42
  datatypes
    Hive complex datatypes 41

H
HDFS connections
  creating 13
  properties 6
HDFS mappings
  data extraction example 15
  description 15
  high availability
    description 2, 39
Hive connections
  creating 13
  properties 7
Hive environment processing
  architecture 3
Hive execution plan
  description, for mapping 19
Hive mappings
  description 16
  workflows 27
Hive query
  description, for mapping 19
  Hive query plan
    viewing, for mapping 28
    viewing, for profile 35
Hive script
  description, for mapping 19

M
mapping example
  Hive 17
  Twitter 18
mapping run on Hadoop
  monitoring 28
  overview 19

N
native environment
  high availability 39
  mappings 14
  optimization 38
  partitioning 38

P
partitioning
  description 2, 38
  optimization 39
PowerCenter Big Data Edition
  overview 1
PowerCenter sessions
  partitioning 38
profile results
  viewing 36
profile run on Hadoop
  monitoring 35
  Overview 30
  profile types 33
  running a single data object 34
  running multiple data objects 35

R
rule profiling on Hadoop
  overview 34

S
social media mappings
  description 17
W

workflows
   Hive mappings 27