Informatica® Big Data Management
10.0

User Guide

Informatica Big Data Management User Guide
10.0
November 2015

© Copyright Informatica LLC 2012, 2018

This software and documentation contain proprietary information of Informatica LLC and are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright law. Reverse engineering of the software is prohibited. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC. This Software may be protected by U.S. and/or international Patents and other Patents Pending.

Use, duplication, or disclosure of the Software by the U.S. Government is subject to the restrictions set forth in the applicable software license agreement and as provided in DFARS 227.7202-1(a) and 227.7702-3(a) (1995), DFARS 252.227-7013©(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable.

The information in this product or documentation is subject to change without notice. If you find any problems in this product or documentation, please report them to us in writing.

Informatica, Informatica Platform, Informatica Data Services, PowerCenter, PowerCenterRT, PowerCenter Connect, PowerCenter Data Analyzer, PowerExchange, PowerMart, Metadata Manager, Informatica Data Quality, Informatica Data Explorer, Informatica B2B Data Transformation, Informatica B2B Data Exchange Informatica On Demand, Informatica Identity Resolution, Informatica Application Information Lifecycle Management, Informatica Complex Event Processing, Ultra Messaging and Informatica Master Data Management are trademarks or registered trademarks of Informatica LLC in the United States and in jurisdictions throughout the world. All other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties, including without limitation: Copyright DataDirect Technologies. All rights reserved. Copyright © Sun Microsystems. All rights reserved. Copyright © RSA Security Inc. All Rights Reserved. Copyright © Ordinal Technology Corp. All rights reserved. Copyright © Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright © Meta Integration Technology, Inc. All rights reserved. Copyright © Intalio. All rights reserved. Copyright © Oracle. All rights reserved. Copyright © Adobe Systems Incorporated. All rights reserved. Copyright © DataArt, Inc. All rights reserved. Copyright © ComponentSource. All rights reserved. Copyright © Microsoft Corporation. All rights reserved. Copyright © Rogue Wave Software, Inc. All rights reserved. Copyright © Teradata Corporation. All rights reserved. Copyright © Yahoo! Inc. All rights reserved. Copyright © Glyph & Cog, LLC. All rights reserved. Copyright © Thinkmap, Inc. All rights reserved. Copyright © Clearpace Software Limited. All rights reserved. Copyright © Information Builders, Inc. All rights reserved. Copyright © OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved. Copyright Cleo Communications, Inc. All rights reserved. Copyright © International Organization for Standardization 1986. All rights reserved. Copyright © ej-technologies GmbH. All rights reserved. Copyright © Jaspersoft Corporation. All rights reserved. Copyright © International Business Machines Corporation. All rights reserved. Copyright © yWorks GmbH. All rights reserved. Copyright © Lucent Technologies. All rights reserved. Copyright (c) University of Toronto. All rights reserved. Copyright © Daniel Veillard. All rights reserved. Copyright © Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright © MicroQuill Software Publishing, Inc. All rights reserved. Copyright © PassMark Software Pty Ltd. All rights reserved. Copyright © LogiXML, Inc. All rights reserved. Copyright © 2003-2010 Lorenzi Davide, All rights reserved. Copyright © Red Hat, Inc. All rights reserved. Copyright © The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright © EMC Corporation. All rights reserved. Copyright © Flexera Software. All rights reserved. Copyright © Jinfonet Software. All rights reserved. Copyright © Apple Inc. All rights reserved. Copyright © Telerik Inc. All rights reserved. Copyright © BEA Systems. All rights reserved. Copyright © PDFlib GmbH. All rights reserved. Copyright © Orientation in Objects GmbH. All rights reserved. Copyright © Tanuki Software, Ltd. All rights reserved. Copyright © Ricebridge. All rights reserved. Copyright © Sencha, Inc. All rights reserved. Copyright © Scalable Systems, Inc. All rights reserved. Copyright © jQWidgets. All rights reserved. Copyright © Tableau Software, Inc. All rights reserved. Copyright© MaxMind, Inc. All Rights Reserved. Copyright © TMate Software s.r.o. All rights reserved. Copyright © MapR Technologies Inc. All rights reserved. Copyright © Amazon Corporate LLC. All rights reserved. Copyright © Highsoft. All rights reserved. Copyright © Python Software Foundation. All rights reserved. Copyright © BeOpen.com. All rights reserved. Copyright © CNRI. All rights reserved.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/), and/or other software which is licensed under various versions of the Apache License (the "License"). You may obtain a copy of these Licenses at http://www.apache.org/licenses/. Unless required by applicable law or agreed to in writing, software distributed under these Licenses is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the Licenses for the specific language governing permissions and limitations under the Licenses.

This product includes software which was developed by Mozilla (http://www.mozilla.org/), software copyright The JBoss Group, LLC, all rights reserved; software copyright © 1999-2006 by Bruno Lowagie and Paulo Soares and other software which is licensed under various versions of the GNU Lesser General Public License Agreement, which may be found at http:// www.gnu.org/licenses/lgpl.html. The materials are provided free of charge by Informatica, "as-is", without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

The product includes ACE(TM) and TAO(TM) software copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University, Copyright (©) 1993-2006, all rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (copyright The OpenSSL Project. All Rights Reserved) and redistribution of this software is subject to terms available at http://www.openssl.org and http://www.openssl.org/source/license.html.

This product includes Curl software which is Copyright 1996-2013, Daniel Stenberg, <daniel@haxx.se>. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http://curl.haxx.se/docs/copyright.html. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

The product includes software copyright 2001-2005 (©) MetaStuff, Ltd. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http://www.dom4j.org/ license.html.

The product includes software copyright © 2004-2007, The Dojo Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http://dojotoolkit.org/license.

This product includes ICU software which is copyright International Business Machines Corporation and others. All rights reserved. Permissions and limitations regarding this software are subject to terms available at http://source.icu-project.org/repos/icu/icu/trunk/license.html.

This product includes software copyright © 1996-2006 Per Bothner. All rights reserved. Your right to use such materials is set forth in the license which may be found at http:// www.gnu.org/software/ kawa/Software-License.html.

This product includes OSSP UUID software which is Copyright © 2002 Ralf S. Engelschall, Copyright © 2002 The OSSP Project Copyright © 2002 Cable & Wireless Deutschland. Permissions and limitations regarding this software are subject to terms available at http://www.opensource.org/licenses/mit-license.php.

This product includes software developed by Boost (http://www.boost.org/) or under the Boost software license. Permissions and limitations regarding this software are subject to terms available at http:/ /www.boost.org/LICENSE_1_0.txt.

This product includes software copyright © 1997-2007 University of Cambridge. Permissions and limitations regarding this software are subject to terms available at http:// www.pcre.org/license.txt.

This product includes software copyright © 2007 The Eclipse Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http:// www.eclipse.org/org/documents/epl-v10.php and at http://www.eclipse.org/org/documents/edl-v10.php.

This product includes software licensed under the terms at http://www.tcl.tk/software/tcltk/license.html, http://www.bosrup.com/web/overlib/?License, http://www.stlport.org/doc/ license.html, http://asm.ow2.org/license.html, http://www.cryptix.org/LICENSE.TXT, http://hsqldb.org/web/hsqlLicense.html, http://httpunit.sourceforge.net/doc/ license.html, http://jung.sourceforge.net/license.txt , http://www.gzip.org/zlib/zlib_license.html, http://www.openldap.org/software/release/license.html, http://www.libssh2.org, http://slf4j.org/license.html, http://www.sente.ch/software/OpenSourceLicense.html, http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3- license-agreement; http://antlr.org/license.html; http://aopalliance.sourceforge.net/; http://www.bouncycastle.org/licence.html; http://www.jgraph.com/jgraphdownload.html; http://www.jcraft.com/jsch/LICENSE.txt; http://jotm.objectweb.org/bsd_license.html; . http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231; http://www.slf4j.org/license.html; http://nanoxml.sourceforge.net/orig/copyright.html; http://www.json.org/license.html; http://forge.ow2.org/projects/javaservice/, http://www.postgresql.org/about/licence.html, http://www.sqlite.org/copyright.html, http://www.tcl.tk/software/tcltk/license.html, http://www.jaxen.org/faq.html, http://www.jdom.org/docs/faq.html, http://www.slf4j.org/license.html; http://www.iodbc.org/dataspace/iodbc/wiki/iODBC/License; http://www.keplerproject.org/md5/license.html; http://www.toedter.com/en/jcalendar/license.html; http://www.edankert.com/bounce/index.html; http://www.net-snmp.org/about/license.html; http://www.openmdx.org/#FAQ; http://www.php.net/license/3_01.txt; http://srp.stanford.edu/license.txt; http://www.schneier.com/blowfish.html; http://www.jmock.org/license.html; http://xsom.java.net; http://benalman.com/about/license/; https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js; http://www.h2database.com/html/license.html#summary; http://jsoncpp.sourceforge.net/LICENSE; http://jdbc.postgresql.org/license.html; http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto; https://github.com/rantav/hector/blob/master/LICENSE; http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html; http://jibx.sourceforge.net/jibx-license.html; https://github.com/lyokato/libgeohash/blob/master/LICENSE; https://github.com/hjiang/jsonxx/blob/master/LICENSE; https://code.google.com/p/lz4/; https://github.com/jedisct1/libsodium/blob/master/LICENSE; http://one-jar.sourceforge.net/index.php?page=documents&file=license; https://github.com/EsotericSoftware/kryo/blob/master/license.txt; http://www.scala-lang.org/license.html; https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt; http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html; https://aws.amazon.com/asl/; https://github.com/twbs/bootstrap/blob/master/LICENSE; https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt; https://github.com/documentcloud/underscore-contrib/blob/master/LICENSE, and https://github.com/apache/hbase/blob/master/LICENSE.txt.

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/bsd-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.

See patents at https://www.informatica.com/legal/patents.html.

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

NOTICES

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

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

Publication Date: 2018-07-03

Table of Contents

Preface

The *Informatica Big Data Management User Guide* provides information about how to configure Informatica products for Hadoop.

Informatica Resources

Informatica My Support Portal

As an Informatica customer, the first step in reaching out to Informatica is through the Informatica My Support Portal at https://mysupport.informatica.com. The My Support Portal is the largest online data integration collaboration platform with over 100,000 Informatica customers and partners worldwide.

As a member, you can:

- Access all of your Informatica resources in one place.
- Review your support cases.
- Search the Knowledge Base, find product documentation, access how-to documents, and watch support videos.
- Find your local Informatica User Group Network and collaborate with your peers.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base, find product documentation, access how-to documents, and watch support videos.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Documentation

The Informatica Documentation team makes 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 https://mysupport.informatica.com.

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. You can access the PAMs on the Informatica My Support Portal at https://mysupport.informatica.com.

Informatica Web Site

You can access the Informatica corporate web site at https://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 https://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 https://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 https://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.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at http://www.informatica.com/us/services-and-training/support-services/global-support-centers/.

CHAPTER 1

Introduction to Informatica Big Data Management

This chapter includes the following topics:

Informatica Big Data Management Overview

Informatica Big Data Management enables your organization to process large, diverse, and fast changing data sets so you can get insights into your data. Use Big Data Management to perform big data integration and transformation without writing or maintaining Apache Hadoop code.

Use Big Data Management to collect diverse data faster, build business logic in a visual environment, and eliminate hand-coding to get insights on your data. Consider implementing a big data project in the following situations:

- The volume of the data that you want to process is greater than 10 terabytes.
- You need to analyze or capture data changes in microseconds.
- The data sources are varied and range from unstructured text to social media data.

When you use Big Data Management to create a big data project, you can identify big data sources and perform profiling to determine the quality of the data. You can build the business logic for the data and push this logic to the Hadoop cluster for faster and more efficient processing. You can view the status of the big data processing jobs and view how the big data queries are performing.

You can use Big Data Management functionality based on your product license. You can use multiple product tools and clients such as Informatica Developer (the Developer tool) and Informatica Administrator (the Administrator tool) to access big data functionality. Big Data Management connects to third-party applications such as the Hadoop Distributed File System (HDFS) and NoSQL databases such as HBase on a Hadoop cluster on different Hadoop distributions.

The Developer tool includes the native and Hadoop run-time environments for optimal processing. The native run-time environment can process data that is less than 10 terabytes. In the native environment, the Data

Integration Service processes the data. The Hadoop run-time environment can optimize mapping performance and process data that is greater than 10 terabytes. In the Hadoop environment, the Data Integration Service pushes the processing to nodes in a Hadoop cluster.

Example

You are an investment banker who needs to calculate the popularity and risk of stocks and then match stocks to each customer based on the preferences of the customer. Your CIO wants to automate the process of calculating the popularity and risk of each stock, match stocks to each customer, and then send an email with a list of stock recommendations for all customers.

You consider the following requirements for your project:

- The volume of data generated by each stock is greater than 10 terabytes.
- You need to analyze the changes to the stock in microseconds.
- The stock is included in Twitter feeds and company stock trade websites, so you need to analyze these social media sources.

Based on your requirements you decide to reach out to your IT department to create a big data project that determines the popularity of a stock. The project counts the number of times the stock is included in Twitter feeds and the number of times customers inquire about the stock on the company stock trade web site.

Big Data Management Tasks

Use Big Data Management when you want to access, analyze, prepare, transform, and stream data faster than traditional data processing environments.

You can use Big Data Management for the following tasks:

- Read from and write to diverse big data sources and targets.
- Perform data replication on a Hadoop cluster.
- Perform data discovery.
- Perform data lineage on big data sources.
- Stream machine data.
- Manage big data relationships.

Note: The *Informatica Big Data Management User Guide* describes how to run big data mappings in the native environment or the Hadoop environment. For information on specific license and configuration requirements for a task, refer to the related product guides.

Read from and Write to Big Data Sources and Targets

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

For more information, see the *Data Transformation HParser Operator Guide*.

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.

For more information about PowerExchange adapters for social media, see the related PowerExchange adapter guides.

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.

For more information about PowerExchange adapters, see the related PowerExchange adapter guides.

Perform Data Discovery

Data discovery is the process of discovering the metadata of source systems that include content, structure, patterns, and data domains. Content refers to data values, frequencies, and data types. Structure includes candidate keys, primary keys, foreign keys, and functional dependencies. The data discovery process offers advanced profiling capabilities.

In the native environment, you can define a profile to analyze data in a single data object or across multiple data objects. In the Hadoop environment, you can push column profiles and the data domain discovery process to the Hadoop cluster.

Run a profile to evaluate the data structure and to verify that data columns contain the types of information you expect. You can drill down on data rows in profiled data. If the profile results reveal problems in the data, you can apply rules to fix the result set. You can create scorecards to track and measure data quality before and after you apply the rules. If the external source metadata of a profile or scorecard changes, you can synchronize the changes with its data object. You can add comments to profiles so that you can track the profiling process effectively.

For more information, see the *Informatica Data Discovery Guide*.

Perform Data Lineage on Big Data Sources

Perform data lineage analysis in Metadata Manager for big data sources and targets.

Use Metadata Manager to create a Cloudera Navigator resource to extract metadata for big data sources and targets and perform data lineage analysis on the metadata. Cloudera Navigator is a data management tool

for the Hadoop platform that enables users to track data access for entities and manage metadata about the entities in a Hadoop cluster.

You can create one Cloudera Navigator resource for each Hadoop cluster that is managed by Cloudera Manager. Metadata Manager extracts metadata about entities from the cluster based on the entity type.

Metadata Manager extracts metadata for the following entity types:

- HDFS files and directories
- Hive tables, query templates, and executions
- Oozie job templates and executions
- Pig tables, scripts, and script executions
- YARN job templates and executions

Note: Metadata Manager does not extract metadata for MapReduce job templates or executions.

For more information, see the *Metadata Manager Administrator Guide*.

Stream Machine Data

You can stream machine data in real time. To stream machine data, use Informatica Vibe Data Stream for Machine Data (Vibe Data Stream).

Vibe Data Stream is a highly available, distributed, real-time application that collects and aggregates machine data. You can collect machine data from different types of sources and write to different types of targets. Vibe Data Stream consists of source services that collect data from sources and target services that aggregate and write data to a target.

For more information, see the *Informatica Vibe Data Stream for Machine Data User Guide*.

Manage Big Data Relationships

You can manage big data relationships by integrating data from different sources and indexing and linking the data in a Hadoop environment. Use Big Data Management to integrate data from different sources. Then use the MDM Big Data Relationship Manager to index and link the data in a Hadoop environment.

MDM Big Data Relationship Manager indexes and links the data based on the indexing and matching rules. You can configure rules based on which to link the input records. MDM Big Data Relationship Manager uses the rules to match the input records and then group all the matched records. MDM Big Data Relationship Manager links all the matched records and creates a cluster for each group of the matched records. You can load the indexed and matched record into a repository.

For more information, see the *MDM Big Data Relationship Management User Guide*.

Big Data Process

To create a big data project, you collect the data from diverse data sources. You can perform profiling, cleansing, and matching for the data. You build the business logic for the data and push the transformed data to the data warehouse. Then you can perform business intelligence on a view of the data.

Based on your big data project, you can perform the following high-level tasks:

1. Collect the data.

2. Cleanse the data
3. Transform the data.
4. Process the data.
5. Monitor jobs.

Step 1. Collect the Data

Identify the data sources from which you need to collect the data.

Big Data Management provides several ways to access your data in and out of Hadoop based on the data types, data volumes, and data latencies in the data.

You can use PowerExchange adapters to connect to multiple big data sources. You can schedule batch loads to move data from multiple source systems to HDFS without the need to stage the data. You can move changed data from relational and mainframe systems into HDFS or the Hive warehouse. For real-time data feeds, you can move data off message queues and into HDFS.

You can collect the following types of data:

- Transactional
- Interactive
- Log file
- Sensor device
- Document and file
- Industry format

Step 2. Cleanse the Data

Cleanse the data by profiling, cleaning, and matching your data. You can view data lineage for the data.

You can perform data profiling to view missing values and descriptive statistics to identify outliers and anomalies in your data. You can view value and pattern frequencies to isolate inconsistencies or unexpected patterns in your data. You can drill down on the inconsistent data to view results across the entire data set.

You can automate the discovery of data domains and relationships between them. You can discover sensitive data such as social security numbers and credit card numbers so that you can mask the data for compliance.

After you are satisfied with the quality of your data, you can also create a business glossary from your data. You can use the Analyst tool or Developer tool to perform data profiling tasks. Use the Analyst tool to perform data discovery tasks. Use Metadata Manager to perform data lineage tasks.

Step 3. Transform the Data

You can build the business logic for your data to parse your data in an easy visual interface. Eliminate the need for hand-coding the transformation logic by using pre-built Informatica transformations to transform your data.

Step 4. Process the Data

Based on your business logic, you can determine the optimal run-time environment to process your data. If your data is less than 10 terabytes, consider processing your data in the native environment. If your data is greater than 10 terabytes, consider processing your data in the Hadoop environment.

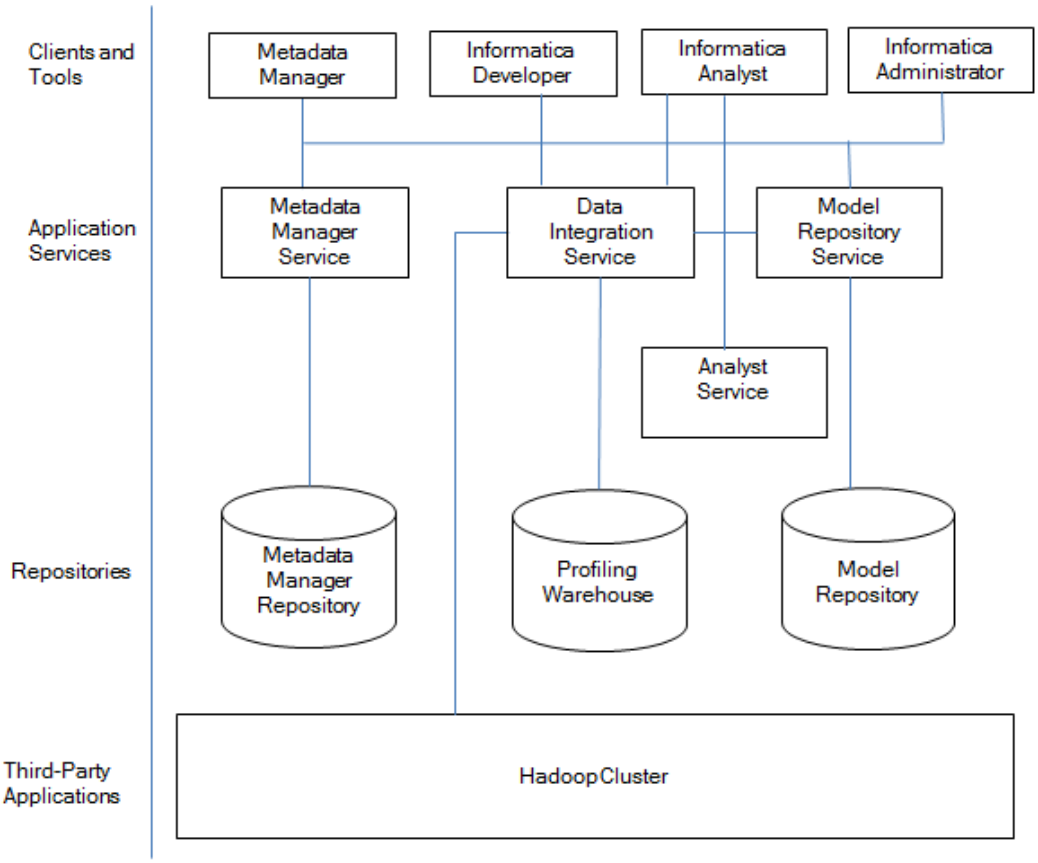
Step 5. Monitor Jobs

Monitor the status of your processing jobs. After your processing jobs complete you can get business intelligence and analytics from your data. You can view monitoring statistics for your processing jobs in the Monitoring tool.

Big Data Management Component Architecture

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

The following image shows the components of Big Data Management:



Clients and Tools

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

Use the following tools to manage big data projects:

Informatica Administrator

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

Informatica Analyst

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

Informatica Developer

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

Application Services

Big Data Management uses application services in the Informatica domain to process data. The application services depend on the task you perform.

Big Data Management uses the following application services:

Analyst Service

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

Data Integration Service

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

Metadata Manager Service

The Metadata Manager Service manages the Metadata Manager repository. The Metadata Manager Service connects to the Metadata Manager repository when you perform data lineage analysis.

You specify a Metadata Manager Service when you configure the Data Integration Service.

Model Repository Service

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

You specify a Model Repository Service when you configure the Data Integration Service.

Repositories

Big Data Management includes repositories to store data related to connections, source metadata, data domains, data profiling, data masking, and data lineage. Big Data Management uses application services in the Informatica domain to access data in repositories. The repositories involved in the process depend on the data store for your task.

Big Data Management includes the following repositories:

Metadata Manager repository

The Metadata Manager Service reads metadata from the Metadata Manager repository when you perform data lineage analysis on the metadata.

Model repository

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

Profiling warehouse

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

Third-Party Applications

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

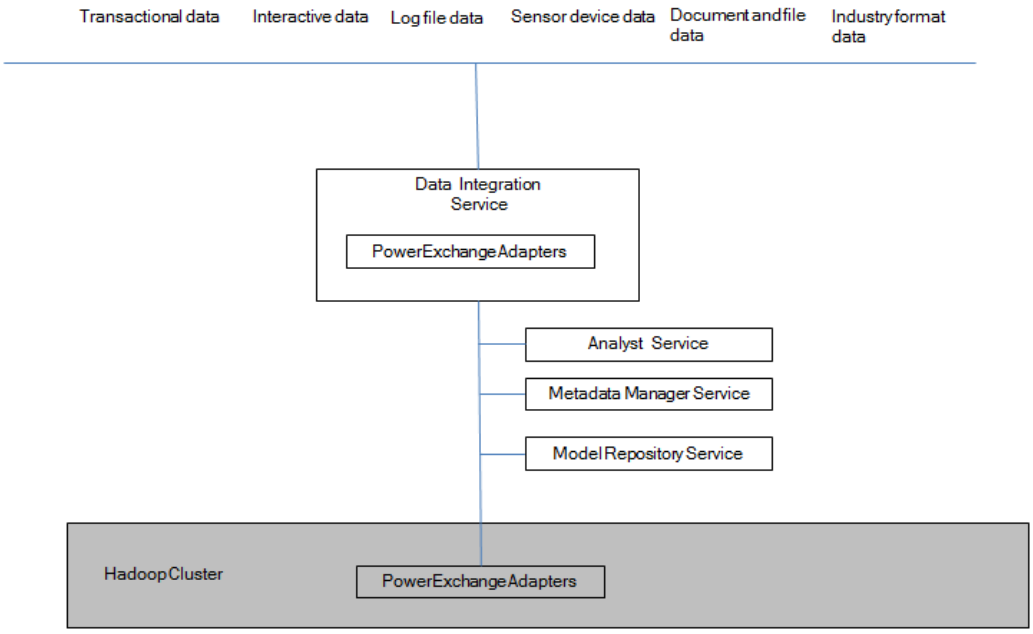
Big Data Management can connect to Hadoop as a data source and push job processing to the Hadoop cluster. It can also connect to HDFS, which enables high performance access to files across the cluster. It can connect to Hive, which is a data warehouse that connects to HDFS and uses SQL-like queries to run MapReduce jobs on Hadoop, or YARN, which can manage Hadoop clusters more efficiently. It can also connect to NoSQL databases such as HBase, which is a database comprising key-value pairs on Hadoop that performs operations in real-time.

Big Data Management Connectivity Architecture

Big Data Management uses PowerExchange adapters to connect to big data sources and uses third-party Hadoop distributions to connect to a Hadoop cluster.

The following image shows the Big Data Management connectivity architecture:

The figure shows the following application services:

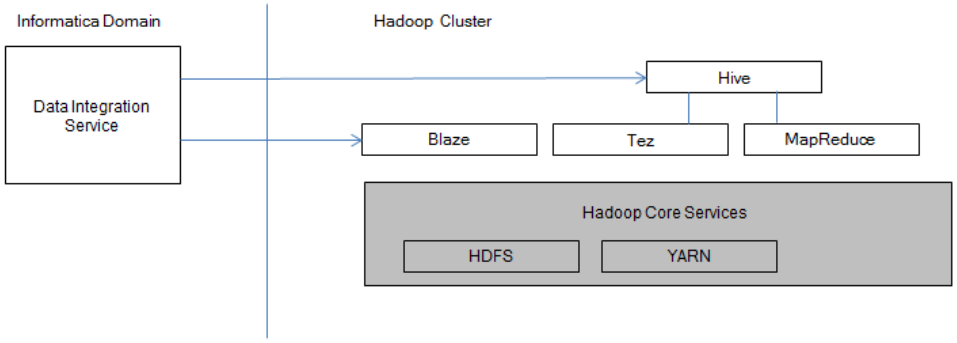


The Data Integration Service uses PowerExchange adapters to connect to big data sources and targets. The Data Integration Service also pushes down job processing to the Hadoop cluster. The Analyst Service, Metadata Manager Service, and Model Repository Service connect to the Data Integration Service.

Hadoop Ecosystem Architecture

The Hadoop ecosystem comprises varied technologies that scale the processing power of Hadoop for big data projects. While it is not possible to list all emerging technologies, Big Data Management interacts with standard Hadoop distributions, HDFS, Hive, and YARN.

The following image shows how Big Data Management interacts with the Hadoop ecosystem:



CHAPTER 2

Connections

This chapter includes the following topics:

Connections Overview

Define the connections that you want to use to access data in HBase, HDFS, or Hive, or run a mapping on a Hadoop cluster. You can create the connections using the Developer tool, Administrator tool, and infacmd.

You can create the following types of connections:

Hadoop connection

Create a Hadoop connection to run mappings on the Hadoop cluster. Select the Hadoop connection if you select the Hadoop run-time environment. You must also select the Hadoop connection to validate a mapping to run on 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.

HDFS connection

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

HBase connection

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

Hive connection

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

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

Hadoop Connection Properties

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

General Properties

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

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

Hadoop Cluster Properties

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

Property	Description

Resource Manager Address	The service within Hadoop that submits requests for resources or spawns YARN applications.
	Use the following format:
	`<hostname>:<port>`
	Where
	- <hostname> is the host name or IP address of the Yarn resource manager.
	- <port> is the port on which the Yarn resource manager listens for remote procedure calls (RPC).
	For example, enter: `myhostame:8032`
	You can also get the Resource Manager Address property from yarn-site.xml located in the following directory on the Hadoop cluster: `/etc/hadoop/conf/`
	The Resource Manager Address appears as the following property in yarn-site.xml:
	`<property>`
	`<name>yarn.resourcemanager.address</name>`
	`<value>hostname:port</value>`
	`<description>The address of the applications manager interface in the Resource Manager.</description>`
	`</property>`
	Optionally, if the `yarn.resourcemanager.address` property is not configured in yarn-site.xml, you can find the host name from the `yarn.resourcemanager.hostname` or `yarn.resourcemanager.scheduler.address` properties in yarn-site.xml. You can then configure the Resource Manager Address in the Hadoop connection with the following value: `hostname:8032`

Default File System URI	The URI to access the default Hadoop Distributed File System.
	Use the following connection URI:
	`hdfs://<node name>:<port>`
	Where
	- <node name> is the host name or IP address of the NameNode.
	- <port> is the port on which the NameNode listens for remote procedure calls (RPC).
	For example, enter: `hdfs://myhostname:8020/`
	You can also get the Default File System URI property from core-site.xml located in the following directory on the Hadoop cluster: `/etc/hadoop/conf/`
	Use the value from the `fs.defaultFS` property found in core-site.xml.
	For example, use the following value:
	`<property>`
	`<name>fs.defaultFS</name>`
	`<value>hdfs://localhost:8020</value>`
	`</property>`
	If the Hadoop cluster runs MapR, use the following URI to access the MapR File system: `maprfs:///`.

Common Properties

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

Property	Description
Impersonation User Name	User name of the user that the Data Integration Service impersonates to run mappings on a Hadoop cluster. If the Hadoop cluster uses Kerberos authentication, the principal name for the JDBC connection string and the user name must be the same. **Note:** You must use user impersonation for the Hadoop connection if the Hadoop cluster uses Kerberos authentication. 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 operating system profile user name of the machine that runs the Data Integration Service.
Temporary Table Compression Codec	Hadoop compression library for a compression codec class name.
Codec Class Name	Codec class name that enables data compression and improves performance on temporary staging tables.
Hadoop Connection Custom Properties	Custom properties that are unique to the Hadoop connection. You can specify multiple properties. Use the following format: `<property1>=<value>` Where - <property1> is a Blaze, Hive, or Hadoop property. - <value> is the value of the Hive or Hadoop property. To specify multiple properties use `&:` as the property separator. Use custom properties only at the request of Informatica Global Customer Support.

Hive Pushdown Configuration

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

Property	Description
Environment SQL	SQL commands to set the Hadoop environment. The Data Integration Service executes the environment SQL at the beginning of each Hive script generated in a Hive execution plan. The following rules and guidelines apply to the usage of environment SQL: - Use the environment SQL to specify Hive queries. - Use the environment SQL to set the classpath for Hive user-defined functions and then use environment SQL or PreSQL to specify the Hive user-defined functions. You cannot use PreSQL in the data object properties to specify the classpath. The path must be the fully qualified path to the JAR files used for user-defined functions. Set the parameter hive.aux.jars.path with all the entries in infapdo.aux.jars.path and the path to the JAR files for user-defined functions. - You can use environment SQL to define Hadoop or Hive parameters that you want to use in the PreSQL commands or in custom queries.
Database Name	Namespace for tables. Use the name `default` for tables that do not have a specified database name.

Hive Warehouse Directory on HDFS	The absolute HDFS file path of the default database for the warehouse that is local to the cluster. For example, the following file path specifies a local warehouse: `/user/hive/warehouse`
	For Cloudera CDH, if the Metastore Execution Mode is remote, then the file path must match the file path specified by the Hive Metastore Service on the Hadoop cluster.
	You can get the value for the Hive Warehouse Directory on HDFS from the `hive.metastore.warehouse.dir` property in hive-site.xml located in the following directory on the Hadoop cluster: `/etc/hadoop/conf/`
	For example, use the following value:
	`<property>`
	`<name>hive.metastore.warehouse.dir</name>`
	`<value>/usr/hive/warehouse </value>`
	`<description>location of the warehouse directory</description>`
	`</property>`
	For MapR, `hive-site.xml` is located in the following direcetory: `/opt/mapr/hive/<hive version>/conf`.

Hive Configuration

You can use the values for Hive configuration properties from hive-site.xml or mapred-site.xml located in the following directory on the Hadoop cluster: `/etc/hadoop/conf/`

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

Property	Description

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, Metastore Database Driver, Username, and Password. For a remote metastore, you must specify only the Remote Metastore URI.
	You can get the value for the Metastore Execution Mode from hive-site.xml. The Metastore Execution Mode appears as the following property in hive-site.xml:
	`<property>`
	`<name>hive.metastore.local</name>`
	`<value>true</true>`
	`</property>`
	Note: The `hive.metastore.local` property is deprecated in hive-site.xml for Hive server versions 0.9 and above. If the `hive.metastore.local` property does not exist but the `hive.metastore.uris` property exists, and you know that the Hive server has started, you can set the connection to a remote metastore.

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`
	You can get the value for the Metastore Database URI from hive-site.xml. The Metastore Database URI appears as the following property in hive-site.xml:
	`<property>`
	`<name>javax.jdo.option.ConnectionURL</name>`
	`<value>jdbc:mysql://MYHOST/metastore</value>`
	`</property>`

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`
	You can get the value for the Metastore Database Driver from hive-site.xml. The Metastore Database Driver appears as the following property in hive-site.xml:
	`<property>`
	`<name>javax.jdo.option.ConnectionDriverName</name>`
	`<value>com.mysql.jdbc.Driver</value>`
	`</property>`

Metastore Database User Name	The metastore database user name.
	You can get the value for the Metastore Database User Name from hive-site.xml. The Metastore Database User Name appears as the following property in hive-site.xml:
	`<property>`
	`<name>javax.jdo.option.ConnectionUserName</name>`
	`<value>hiveuser</value>`
	`</property>`

Property	Description

Metastore Database Password	The password for the metastore user name.
	You can get the value for the Metastore Database Password from hive-site.xml. The Metastore Database Password appears as the following property in hive-site.xml:
	`<property>`
	`  <name>javax.jdo.option.ConnectionPassword</name>`
	`  <value>password</value>`
	`</property>`

Remote Metastore URI	The metastore URI used to access metadata in a remote metastore setup. For a remote metastore, you must specify the Thrift server details.
	Use the following connection URI:
	`thrift://<hostname>:<port>`
	Where
	- <hostname> is name or IP address of the Thrift metastore server.
	- <port> is the port on which the Thrift server is listening.
	For example, enter: `thrift://myhostname:9083/`
	You can get the value for the Remote Metastore URI from hive-site.xml. The Remote Metastore URI appears as the following property in hive-site.xml:
	`<property>`
	`  <name>hive.metastore.uris</name>`
	`  <value>thrift://<n.n.n.n>:9083</value>`
	`<description> IP address or fully-qualified domain name and port of the metastore host</description>`
	`</property>`

Property	Description

Engine Type	The engine that the Hadoop environment uses to run a mapping on the Hadoop cluster. Select a value from the drop down list.
	For example select: `MRv2`
	To set the engine type in the Hadoop connection, you must get the value for the `mapreduce.framework.name` property from mapred-site.xml located in the following directory on the Hadoop cluster: `/etc/hadoop/conf/`
	If the value for `mapreduce.framework.name` is `classic`, select `mrv1` as the engine type in the Hadoop connection.
	If the value for `mapreduce.framework.name` is `yarn`, you can select the `mrv2` or `tez` as the engine type in the Hadoop connection. Do not select Tez if Tez is not configured for the Hadoop cluster.
	You can also set the value for the engine type in hive-site.xml. The engine type appears as the following property in hive-site.xml:
	`<property>`
	`<name>hive.execution.engine</name>`
	`<value>tez</value>`
	`<description>Chooses execution engine. Options are: mr (MapReduce, default) or tez (Hadoop 2 only)</description>`
	`</property>`

Job Monitoring URL	The URL for the MapReduce JobHistory server. You can use the URL for the JobTracker URI if you use MapReduce version 1.
	Use the following format:
	`<hostname>:<port>`
	Where
	- <hostname> is the host name or IP address of the JobHistory server.
	- <port> is the port on which the JobHistory server listens for remote procedure calls (RPC).
	For example, enter: `myhostname:8021`
	You can get the value for the Job Monitoring URL from mapred-site.xml. The Job Monitoring URL appears as the following property in mapred-site.xml:
	`<property>`
	`<name>mapred.job.tracker</name>`
	`<value>myhostname:8021 </value>`
	`<description>The host and port that the MapReduce job tracker runs at.</description>`
	`</property>`

Blaze Service

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

Property	Description
Temporary Working Directory on HDFS	The HDFS file path of the directory that the Blaze engine uses to store temporary files. Verify that the directory exists. The YARN user, Blaze engine user, and mapping impersonation user must have write permission on this directory. For example, enter: `/blaze/workdir`
Blaze Service User Name	The operating system profile user name for the Blaze engine.
Minimum Port	The minimum value for the port number range for the Blaze engine. For example, enter: `12300`

Property	Description
Maximum Port	The maximum value for the port number range for the Blaze engine. For example, enter: `12600`
Yarn Queue Name	The YARN scheduler queue name used by the Blaze engine that specifies available resources on a cluster. The name is case sensitive.
Blaze Service Custom Properties	Custom properties that are unique to the Blaze engine. You can specify multiple properties. Use the following format: `<property1>=<value>` Where - <property1> is a Blaze engine optimization property. - <value> is the value of the Blaze engine optimization property. To specify multiple properties use `&:` as the property separator. Use custom properties only at the request of Informatica Global Customer Support.

HDFS Connection Properties

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

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

The following table describes HDFS connection properties:

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

Property	Description
User Name	User name to access HDFS.
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 the following for the NameNode URI for MapR clusters: - `maprfs:///`

HBase Connection Properties

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

The following table describes HBase connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: `~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /`
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select HBase.
ZooKeeper Host(s)	Name of the machine that hosts the ZooKeeper server. When the ZooKeeper runs in the replicated mode, specify a comma-separated list of servers in the ZooKeeper quorum servers. If the TCP connection to the server breaks, the client connects to a different server in the quorum.
ZooKeeper Port	Port number of the machine that hosts the ZooKeeper server. If the Hadoop cluster uses MapR, use the value specified for hbase.zookeeper.property.clientPort in hbase-site.xml. You can find `hbase-site.xml` on the NameNode machine in the following directory: `/opt/mapr/hbase/hbase-0.98.7/conf`.

Property	Description
Enable Kerberos Connection	Enables the Informatica domain to communicate with the HBase master server or region server that uses Kerberos authentication.
HBase Master Principal	Service Principal Name (SPN) of the HBase master server. Enables the ZooKeeper server to communicate with an HBase master server that uses Kerberos authentication. Enter a string in the following format: `hbase/<domain.name>@<YOUR-REALM>` Where: - domain.name is the domain name of the machine that hosts the HBase master server. - YOUR-REALM is the Kerberos realm.
HBase Region Server Principal	Service Principal Name (SPN) of the HBase region server. Enables the ZooKeeper server to communicate with an HBase region server that uses Kerberos authentication. Enter a string in the following format: `hbase_rs/<domain.name>@<YOUR-REALM>` Where: - domain.name is the domain name of the machine that hosts the HBase master server. - YOUR-REALM is the Kerberos realm.

Hive Connection Properties

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

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

The following table describes Hive connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: `~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /`
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 4000 characters.

Property	Description
Location	The domain where you want to create the connection. Not valid for the Analyst tool.
Type	The connection type. Select Hive.
Connection Modes	Hive connection mode. Select 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 must 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 profiles in the Hadoop cluster. You can select both the options. Default is **Access Hive as a source or target**.
User Name	User name of the user that the Data Integration Service impersonates to run mappings on a Hadoop cluster. The user name depends on the JDBC connection string that you specify in the Metadata Connection String or Data Access Connection String for the native environment. If the Hadoop cluster uses Kerberos authentication, the principal name for the JDBC connection string and the user name must be the same. Otherwise, the user name depends on the behavior of the JDBC driver. With Hive JDBC driver, you can specify a user name in many ways and the user name can become a part of the JDBC URL. If the Hadoop cluster does not use Kerberos authentication, the user name depends on the behavior of the JDBC driver. If you do not specify a user name, the Hadoop cluster authenticates jobs based on the following criteria: - The Hadoop cluster does not use Kerberos authentication. It authenticates jobs based on the operating system profile user name of the machine that runs the Data Integration Service. - The Hadoop cluster uses Kerberos authentication. It authenticates jobs based on the SPN of the Data Integration Service.
Common Attributes to Both the Modes: Environment SQL	SQL commands to set the Hadoop environment. In native environment type, the Data Integration Service executes the environment SQL each time it creates a connection to a Hive metastore. If you use the Hive connection to run profiles 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 connection modes: - Use the environment SQL to specify Hive queries. - Use the environment SQL to set the classpath for Hive user-defined functions and then use environment SQL or PreSQL to specify the Hive user-defined functions. You cannot use PreSQL in the data object properties to specify the classpath. The path must be the fully qualified path to the JAR files used for user-defined functions. Set the parameter hive.aux.jars.path with all the entries in infapdo.aux.jars.path and the path to the JAR files for user-defined functions. - You can use environment SQL to define Hadoop or Hive parameters that you want to use in the PreSQL commands or in custom queries. If you use the Hive connection to run profiles in the Hadoop cluster, the Data Integration service executes only the environment SQL of the Hive connection. If the Hive sources and targets are on different clusters, the Data Integration Service does not execute the different environment SQL commands for the connections of the Hive source or target.

Properties to Access Hive as Source or Target

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

Property	Description
Metadata Connection String	The JDBC connection URI used to access the metadata from the Hadoop server. You can use PowerExchange for Hive to communicate with a HiveServer service or HiveServer2 service. To connect to HiveServer, specify the connection string in the following format: `jdbc:hive2://<hostname>:<port>/<db>` Where - <hostname> is name or IP address of the machine on which HiveServer2 runs. - <port> is the port number on which HiveServer2 listens. - <db> is the database name to which you want to connect. If you do not provide the database name, the Data Integration Service uses the default database details. To connect to HiveServer 2, use the connection string format that Apache Hive implements for that specific Hadoop Distribution. For more information about Apache Hive connection string formats, see the Apache Hive documentation.
Bypass Hive JDBC Server	JDBC driver mode. Select the check box to use the embedded JDBC driver mode. To use the JDBC embedded mode, perform the following tasks: - Verify that Hive client and Informatica services are installed on the same machine. - Configure the Hive connection properties to run mappings in the Hadoop cluster. If you choose the non-embedded mode, you must configure the Data Access Connection String. Informatica recommends that you use the JDBC embedded mode.
Data Access Connection String	The connection string to access data from the Hadoop data store. To connect to HiveServer, specify the non-embedded JDBC mode connection string in the following format: `jdbc:hive2://<hostname>:<port>/<db>` Where - <hostname> is name or IP address of the machine on which HiveServer2 runs. - <port> is the port number on which HiveServer2 listens. - <db> is the database to which you want to connect. If you do not provide the database name, the Data Integration Service uses the default database details. To connect to HiveServer 2, use the connection string format that Apache Hive implements for the specific Hadoop Distribution. For more information about Apache Hive connection string formats, see the Apache Hive documentation.

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:

Property	Description
Database Name	Namespace for tables. Use the name `default` for tables that do not have a specified database name.
Default FS URI	The URI to access the default Hadoop Distributed File System. Use the following connection URI: `hdfs://<node name>:<port>` Where - <node name> is the host name or IP address of the NameNode. - <port> is the port on which the NameNode listens for remote procedure calls (RPC). If the Hadoop cluster runs MapR, use the following URI to access the MapR File system: `maprfs:///`.
JobTracker/Yarn Resource Manager URI	The service within Hadoop that submits the MapReduce tasks to specific nodes in the cluster. Use the following format: `<hostname>:<port>` Where - <hostname> is the host name or IP address of the JobTracker or Yarn resource manager. - <port> is the port on which the JobTracker or Yarn resource manager listens for remote procedure calls (RPC). If the cluster uses MapR with YARN, use the value specified in the `yarn.resourcemanager.address` property in yarn-site.xml. You can find `yarn-site.xml` in the following directory on the NameNode of the cluster: `/opt/mapr/hadoop/hadoop-2.5.1/etc/hadoop`. MapR with MapReduce 1 supports a highly available JobTracker. If you are using MapR distribution, define the JobTracker URI in the following format: `maprfs:///`
Hive Warehouse Directory on HDFS	The absolute HDFS file path of the default database for the warehouse that is local to the cluster. For example, the following file path specifies a local warehouse: `/user/hive/warehouse` For Cloudera CDH, if the Metastore Execution Mode is remote, then the file path must match the file path specified by the Hive Metastore Service on the Hadoop cluster. For MapR, use the value specified for the `hive.metastore.warehouse.dir` property in `hive-site.xml`. You can find `hive-site.xml` in the following directory on the node that runs HiveServer2: `/opt/mapr/hive/hive-0.13/conf`.

Property	Description

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

Temporary Table Compression Codec	Hadoop compression library for a compression codec class name.
Codec Class Name	Codec class name that enables data compression and improves performance on temporary staging tables.
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`
	For MapR, use the value specified for the `javax.jdo.option.ConnectionURL` property in `hive-site.xml`. You can find hive-site.xml in the following directory on the node where HiveServer 2 runs: /opt/mapr/hive/hive-0.13/conf.

Property	Description
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` For MapR, use the value specified for the `javax.jdo.option.ConnectionDriverName` property in `hive-site.xml`. You can find `hive-site.xml` in the following directory on the node where HiveServer 2 runs: `/opt/mapr/hive/hive-0.13/conf`.
Metastore Database Username	The metastore database user name. For MapR, use the value specified for the `javax.jdo.option.ConnectionUserName` property in `hive-site.xml`. You can find `hive-site.xml` in the following directory on the node where HiveServer 2 runs: `/opt/mapr/hive/hive-0.13/conf`.
Metastore Database Password	The password for the metastore user name. For MapR, use the value specified for the `javax.jdo.option.ConnectionPassword` property in `hive-site.xml`. You can find `hive-site.xml` in the following directory on the node where HiveServer 2 runs: `/opt/mapr/hive/hive-0.13/conf`.
Remote Metastore URI	The metastore URI used to access metadata in a remote metastore setup. For a remote metastore, you must specify the Thrift server details. Use the following connection URI: `thrift://<hostname>:<port>` Where - <hostname> is name or IP address of the Thrift metastore server. - <port> is the port on which the Thrift server is listening. For MapR, use the value specified for the `hive.metastore.uris` property in `hive-site.xml`. You can find `hive-site.xml` in the following directory on the node where HiveServer 2 runs: `/opt/mapr/hive/hive-0.13/conf`.

Creating a Connection to Access Sources or Targets

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

1. Click **Window** > **Preferences**.
2. Select **Informatica** > **Connections**.
3. Expand the domain in the **Available Connections** list.
4. Select the type of connection that you want to create:
   - To select an HBase connection, select **NoSQL** > **HBase**.
   - To select an HDFS connection, select **File Systems** > **Hadoop File System**.
   - To select a Hive connection, select **Database** > **Hive**.
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:

Option	Description
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 profiles 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**.

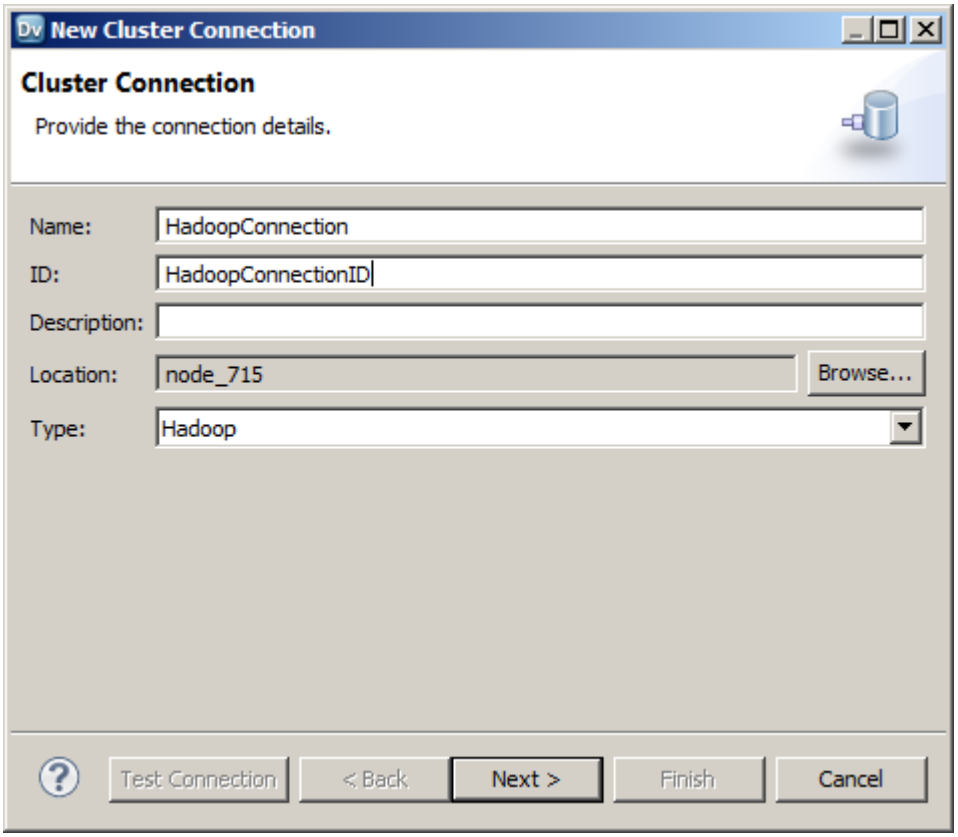
Creating a Hadoop Connection

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

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

   The **New Cluster Connection** dialog box appears.

5. Enter the general properties for the connection.



6. Click **Next**.
7. Enter the Hadoop cluster properties and the common properties for the Hadoop connection.
8. Click **Next**.
9. Enter the Hive pushdown configuration properties and the Hive configuration.
10. Click **Next**.
11. Enter the properties for the Blaze engine.
12. Click **Finish**.

CHAPTER 3

Mappings in a Hadoop Environment

This chapter includes the following topics:

Mappings in a Hadoop Environment Overview

Use the Hadoop run-time environment in the Developer tool to optimize mapping performance and process data that is greater than 10 terabytes. In the Hadoop environment, the Data Integration Service pushes the processing to nodes on a Hadoop cluster. When you select the Hadoop environment, you can also select the engine to push the mapping logic to the Hadoop cluster.

You can run standalone mappings, mappings that are a part of a workflow in the Hadoop environment.

Based on the mapping logic, the Hadoop environment can use the following engines to push processing to nodes on a Hadoop cluster:

- Hive engine. The Hive engine uses Hadoop technology such as MapReduce or Tez for processing batch data.
- Informatica Blaze engine. The Blaze engine is an Informatica proprietary engine for distributed processing on Hadoop.

You can select which engine the Data Integration Service uses.

When you run a mapping in the Hadoop environment, you must configure a Hadoop connection for the mapping. When you edit the Hadoop connection, you can view or configure run-time properties for the Hadoop environment. You can configure the Hive and Blaze engine properties in the Hadoop connection. You can also use parameters to represent properties in the Hadoop environment if you need to use constant values between mapping runs.

You can view the execution plan for a mapping in the Hadoop environment. Viewing the execution plan might enable you to tune the mapping to improve performance. The Hadoop execution plan displays the execution plan for the engine that the Data Integration Service selects to run the mapping.

When you run the mapping, the Data Integration Service converts the mapping to a Hive or Blaze engine execution plan that runs on a Hadoop cluster. You can view the Hive or Blaze engine execution plan using the Developer tool or the Administrator tool.

You can monitor Hive queries and the Hadoop jobs associated with a query for a Hive engine mapping in the Monitoring tool. You can also monitor Blaze engine mapping jobs in the Monitoring tool, or monitor the jobs on a Hadoop cluster with the Blaze Job Monitor web application.

The Data Integration Service logs messages from the DTM, Blaze engine, and Hive engine in the runtime log files.

Data Warehouse Optimization Mapping Example

You can optimize an enterprise data warehouse with the Hadoop system to store more terabytes of data cheaply in the warehouse.

For example, you need to analyze customer portfolios by processing the records that have changed in a 24-hour time period. You can offload the data on Hadoop, find the customer records that have been inserted, deleted, and updated in the last 24 hours, and then update those records in your data warehouse. You can capture these changes even if the number of columns change or if the keys change in the source files.

To capture the changes, you can create the following mappings in the Developer tool:

Mapping_Day1

Create a mapping to read customer data from flat files in a local file system and write to an HDFS target for the first 24-hour period.

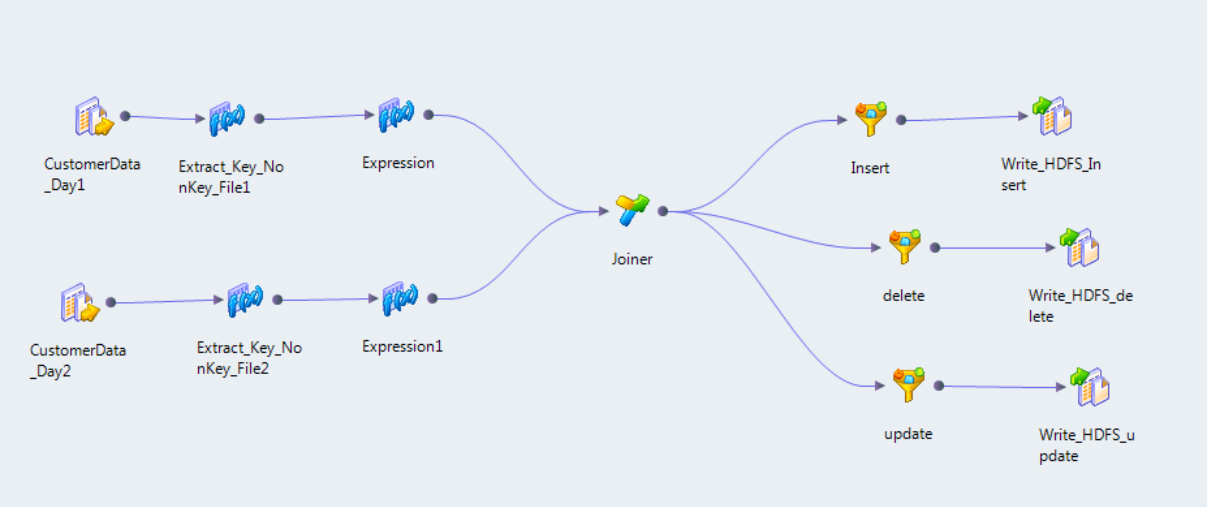
Mapping_Day2

Create a mapping to read customer data from flat files in a local file system and write to an HDFS target for the next 24-hour period.

m_CDC_DWHOptimization

Create a mapping to capture the changed data. The mapping reads data from HDFS and identifies the data that has changed. To increase performance, you configure the mapping to run on Hadoop cluster nodes in a Hadoop environment.

The following image shows the mapping m_CDC_DWHOptimization:



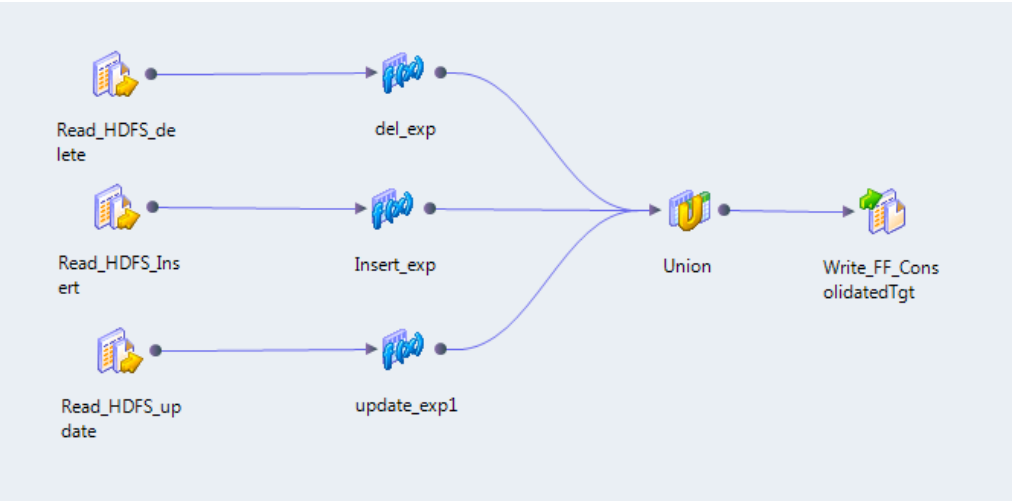
The mapping contains the following objects:

- Read transformations. Transformations that read data from HDFS files that were the targets of Mapping_Day1 and Mapping_Day2. The Data Integration Service reads all of the data as a single column.
- Expression transformations. Extract a key from the non-key values in the data. The expressions use the INSTR function and SUBSTR function to perform the extraction of key values.
- Joiner transformation. Performs a full outer join on the two sources based on the keys generated by the Expression transformations.
- Filter transformations. Use the output of the Joiner transformation to filter rows based on whether or not the rows should be updated, deleted, or inserted.
- Write transformations. Transformations that write the data to three HDFS files based on whether the data is inserted, deleted, or updated.

Consolidated_Mapping

Create a mapping to consolidate the data in the HDFS files and load the data to the data warehouse.

The following figure shows the mapping Consolidated_Mapping:

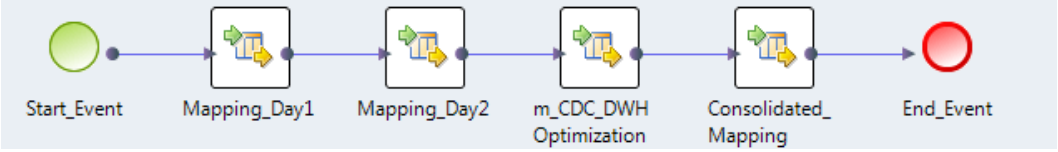


The mapping contains the following objects:

- Read transformations. Transformations that read data from HDFS files that were the target of the previous mapping are the sources of this mapping.
- Expression transformations. Add the deleted, updated, or inserted tags to the data rows.
- Union transformation. Combines the records.
- Write transformation. Transformation that writes data to the flat file that acts as a staging location on the local file system.

You can open each mapping and right-click to run the mapping. To run all mappings in sequence, use a workflow.

The following image shows the example Data Warehouse Optimization workflow:



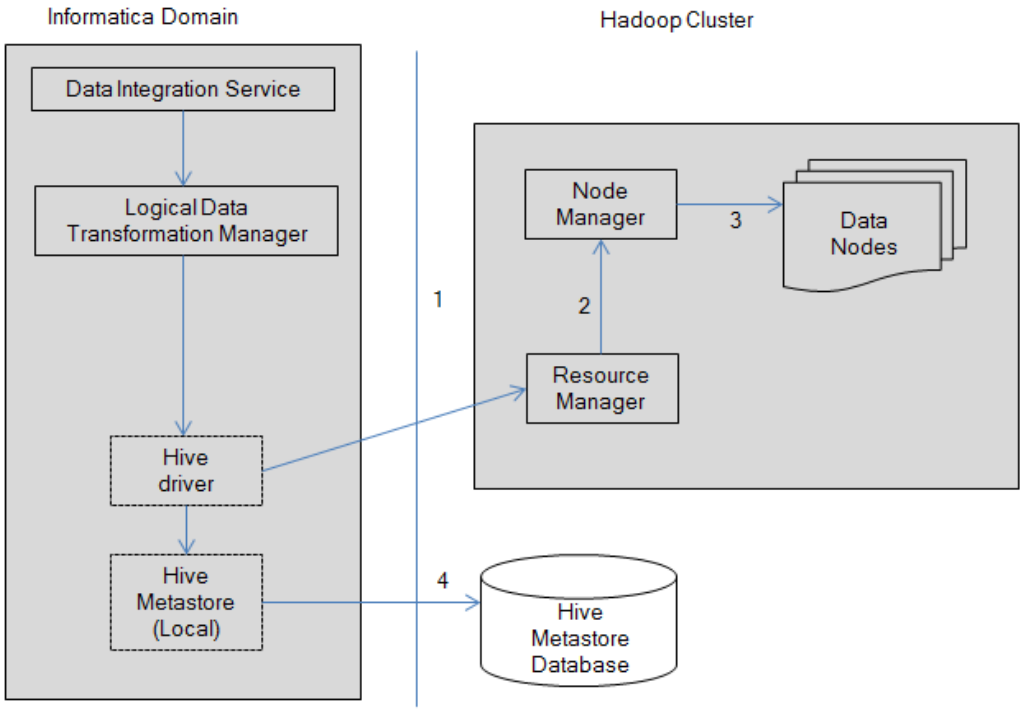
To run the workflow, use the infacmd wfs startWorkflow command.

Hive Engine Architecture

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

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

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



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

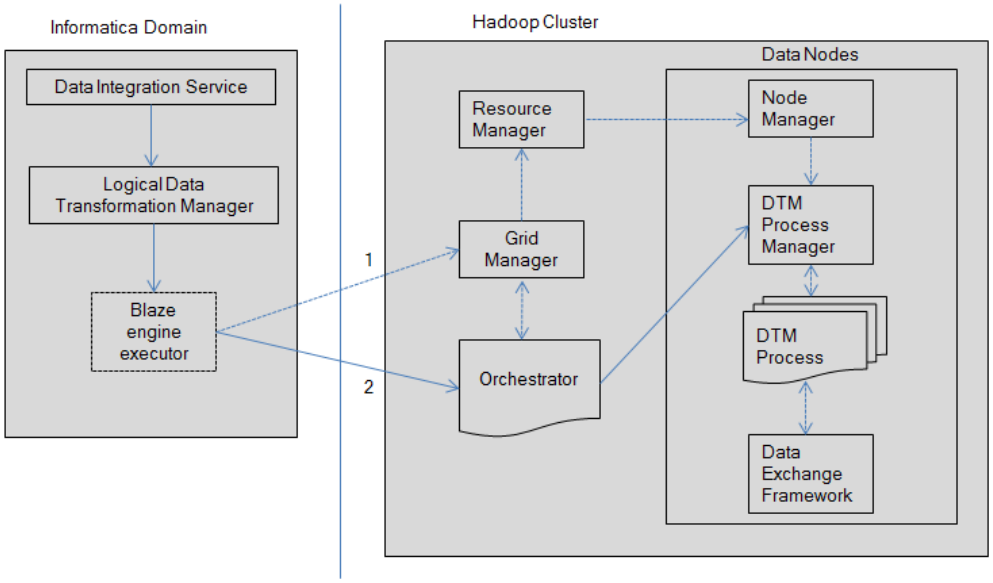
1. The Hive driver sends the MapReduce jobs to the Resource Manager in the Hadoop cluster.
2. The Resource Manager sends the jobs request to the Node Manager that retrieves a list of data nodes that can process the MapReduce jobs.
3. The Node Manager assigns MapReduce jobs to the data nodes.
4. The Hive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

Informatica Blaze Engine Architecture

When you select the Blaze engine the Data Integration Service uses the Blaze engine to run Model repository mappings on the Hadoop cluster.

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

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



The following Blaze engine components appear on the Hadoop cluster:

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

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

1. Initializing Blaze engine components

When the Blaze engine executor receives a job request from the LDTM, it initializes the communication with the Grid Manager to trigger initialization of Blaze engine components on the Hadoop cluster.

2. Executing jobs

The Blaze engine executor executes the mapping job by querying the Grid Manager for an available Orchestrator. The Orchestrator communicates with the Grid Manager and the Resource Manager for available resources on the Hadoop cluster. The Resource Manager sends a job request to the Node Manager on the Hadoop cluster. The Node Manager sends the tasks to the DTM Process Manager, which sends the tasks to the DTM Processes for processing. The DTM Processes communicate with the Data Exchange Framework to send and receive data across processing units that run on the cluster nodes.

High-Level Steps to Run a Mapping in the Hadoop Environment

When you run a mapping in the Hadoop environment, you must configure a Hadoop connection for the mapping. Validate the mapping to ensure that you can push the mapping logic to Hadoop. After you validate a mapping for the Hadoop environment, you can run the mapping.

The following high-level steps describe how to run a mapping in the Hadoop environment:

1. In the Developer tool, create a Hadoop connection.
2. Create a mapping in the Developer tool.
3. Configure the mapping to run in a Hadoop environment.
4. Validate the mapping.
5. Optionally, include the mapping in a workflow.
6. Run the mapping or workflow.

Sources in a Hadoop Environment

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

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

- IBM DB2
- Flat file
- HBase
- HDFS complex file
- HDFS flat file
- Hive
- ODBC
- Oracle
- Teradata
- Netezza

Note: The Blaze engine might not support all sources in a Hadoop environment. When the Data Integration Service finds a source that is not supported by the Blaze engine, it defaults to the Hive engine and runs the mapping on the Hadoop cluster.

Flat File Sources

Mappings that read from flat file sources are valid in a Hadoop environment with some restrictions. A mapping that reads from a flat file source can fail to run in certain cases.

Mappings that read from flat file sources are valid in a Hadoop 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 Hadoop environment with some restrictions.

Hive sources are valid in mappings in a Hadoop 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 Hadoop 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.
- When you run a mapping with a Hive source in the native environment, the Data Integration Service converts decimal values with precision greater than 28 digits to double values.
- When you import Hive tables with Decimal data type into the Developer tool, the Decimal data type precision is set as 38 and the scale is set as 0. This occurs because the third-party Hive JDBC driver does not return the correct precision and scale values for the Decimal data type. For Hive tables on Hive 0.11 and Hive 0.12, accept the default precision and scale for the Decimal data type set by the Developer tool. For Hive tables on Hive 0.12 with Cloudera CDH 5.0, and Hive 0.13 and above, you can configure the precision and scale fields for source columns with the Decimal data type in the Developer tool. For Hive tables on Hive 0.14 or above, the precision and scale used for the Decimal data type in the Hive database also appears in the Developer tool.

Relational Sources

Relational sources are valid in mappings in a Hadoop 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 Hadoop 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 Hadoop environment.

Targets in a Hadoop Environment

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

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

- Greenplum
- IBM DB2
- Flat file
- HBase
- HDFS flat file
- Hive
- ODBC
- Oracle
- Teradata
- Netezza

Note: The Blaze engine might not support all targets in a Hadoop environment. When the Data Integration Service finds a target that is not supported by the Blaze engine, it defaults to the Hive engine and runs the mapping on the Hadoop cluster.

Flat File Targets

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

Flat file targets are valid in mappings in a Hadoop 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 Targets

HDFS flat file targets are valid in mappings in a Hadoop 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 Hadoop environment.

Hive Targets

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

Hive targets are valid in mappings in a Hadoop 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.
- The Data Integration Service can truncate the partition in the Hive target in which the data is being inserted. You must choose to both truncate the partition in the Hive target and truncate the target table.

Relational Targets

Relational targets are valid in mappings in a Hadoop 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 Hadoop environment. You cannot validate and run a mapping with PreSQL or PostSQL properties for a relational target in a Hadoop environment.

The Data Integration Service can use multiple partitions to write to the following relational targets:

- IBM DB2
- Oracle

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

Transformations in a Hadoop Environment

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

Note: The Blaze engine might not support all transformations in a Hadoop environment. When the Data Integration Service finds a transformation that is not supported by the Blaze engine, it defaults to the Hive engine and runs the mapping on the Hadoop cluster.

The following table describes the rules and guidelines for transformations:

Transformation	Rules and Guidelines
Address Validator	You can push mapping logic that includes an Address Validator transformation to Hadoop if you use a Data Quality product license. The following limitation applies to Address Validator transformations: - An Address Validator transformation does not generate a certification report when it runs in a mapping on Hadoop. If you select a certification report option on the transformation, the mapping validation fails when you attempt to push transformation logic to Hadoop.
Aggregator	An Aggregator transformation with pass-through fields is valid if they are group-by fields. You can use the ANY function in an Aggregator transformation with pass-through fields to return any row.

Transformation	Rules and Guidelines
Case Converter	The Data Integration Service can push a Case Converter transformation to Hadoop.
Comparison	You can push mapping logic that includes a Comparison transformation to Hadoop if you use a Data Quality product license.
Consolidation	You can push mapping logic that includes a Consolidation transformation to Hadoop if you use a Data Quality product license. The following limitation applies to Consolidation transformations: - A Consolidation transformation may process records in a different order in native and Hadoop environments. The transformation may identify a different record as the survivor record in each environment.
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. - Pass-through ports cannot be used. - Additional input ports cannot be used. The following limitations apply when a mapping has a Data Processor transformation: - Ports cannot be defined as file. - Ports cannot be defined as binary - Streamer cannot be defined as startup component. The Data Processor transformation can use the following input and output formats: - ASN.1 - Avro - Cobol - JSON - Parquet - XML
Decision	You can push mapping logic that includes a Decision transformation to Hadoop if you use a Data Quality product license.
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.

Transformation	Rules and Guidelines
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 Hadoop 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 or parameters in the outer join condition.
Key Generator	You can push mapping logic that includes a Key Generator transformation to Hadoop if you use a Data Quality product license.
Labeler	You can push mapping logic that includes a Labeler transformation to Hadoop when you configure the transformation to use probabilistic matching techniques. You can push mapping logic that includes all types of Labeler configuration if you use a Data Quality product license.
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.
Match	You can push mapping logic that includes a Match transformation to Hadoop if you use a Data Quality product license. The following limitation applies to Match transformations: - A Match transformation generates cluster ID values differently in native and Hadoop environments. In a Hadoop environment, the transformation appends a group ID value to the cluster ID.
Merge	The Data Integration Service can push a Merge transformation to Hadoop.
Parser	You can push mapping logic that includes a Parser transformation to Hadoop when you configure the transformation to use probabilistic matching techniques. You can push mapping logic that includes all types of Parser configuration if you use a Data Quality product license.
Rank	A comparison is valid if it is case sensitive.

Transformation	Rules and Guidelines
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.
Standardizer	You can push mapping logic that includes a Standardizer transformation to Hadoop if you use a Data Quality product license.
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.
Weighted Average	You can push mapping logic that includes a Weighted Average transformation to Hadoop if you use a Data Quality product license.

Variable Ports in a Hadoop Environment

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

A stateful variable port refers to values from previous rows.

Functions in a Hadoop Environment

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

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

Name	Limitation
ABORT	String argument is not valid.
CUME	Not valid
ERROR	String argument is not valid.
FIRST	Not valid. Use the ANY function to get any row.
LAST	Not valid. Use the ANY function to get any row.
MAX (Dates)	Not valid
MIN (Dates)	Not valid
MOVINGAVG	Not valid

Name	Limitation
MOVINGSUM	Not valid
UUID4()	Valid as an argument in UUID_UNPARSE or ENC_BASE64.
UUID_UNPARSE(Binary)	Valid if the argument is UUID4().

Note: The Hadoop environment treats "/n" values as null values. If an aggregate function contains empty or NULL values, the Hadoop environment includes these values while performing an aggregate calculation.

Mappings in a Hadoop Environment

You can run mappings in a Hadoop environment. When you run mappings in a Hadoop environment, some differences in processing and configuration apply.

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

- A mapping is run in high precision mode in a Hadoop environment for Hive 0.11 and above.
- In a Hadoop 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.
- When you set a limit on the number of rows read from the source for a Blaze mapping, the Data Integration Service runs the mapping with the Hive engine instead of the Blaze engine.

The following configuration differences apply to mappings in a Hadoop 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`.

Data Types in a Hadoop Environment

When you push data types to a Hadoop environment, some variations apply in the processing and validity of data types because of differences between the environments.

The following variations apply in data type processing and validity:

- If a transformation in a mapping has a port with a Binary data type, you can validate and run the mapping in a Hadoop environment.
- You can use high precision Decimal data type with Hive 0.11 and above. When you run mappings in a Hadoop environment, the Data Integration Service converts decimal values with a precision greater than 38 digits to double values. When you run mappings that do not have high precision enabled, the Data Integration Service converts decimal values to double values.
- When you run a mapping with a Hive target that uses the Double data type, the Data Integration Service processes the double data up to 17 digits after the decimal point.
- 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 Hadoop environment.
- Hive complex data types 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 data type and writes to a flat file target, the results can differ between the native environment and a Hadoop environment. For example, in a Hadoop environment, HDFS writes the output string for the decimal 19711025 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 data types when there is data overflow during data type conversion. Mapping results can vary between the native and Hadoop environment when there is data overflow during data type conversion for BigInt and Integer data types.

Parameters in a Hadoop Environment

A mapping parameter represents a constant value that you can change between mapping runs. Use parameters to change the values of connections, file directories, expression components, port lists, port links, and task properties. You can use system parameters or user-defined parameters.

System parameters are built-in parameters for a Data Integration Service. System parameters define the directories where the Data Integration Service stores log files, cache files, reject files, source files, target files, and temporary files. An administrator defines the system parameter default values for a Data Integration Service in the Administrator tool.

User-defined parameters are parameters that you define in transformations, mappings, or workflows. Create user-defined parameters to rerun a mapping with different connection, flat file, cache file, temporary file, expression, ports, or reference table values.

You can override parameter values using a parameter set or a parameter file. A parameter set is a repository object that contains mapping parameter values. A parameter file is an XML file that contains parameter values. When you run the mapping with a parameter set or a parameter file, the Data Integration Service uses the parameter values defined in the parameter set or parameter file instead of the default parameter values you configured in the transformation, mapping, or workflow.

You can use the following parameters to represent additional properties in the Hadoop environment:

Parameters for sources and targets

You can use parameters to represent additional properties for the following big data sources and targets:

- Complex file
- Flat file
- HBase
- HDFS
- Hive

Parameters for the Hadoop connection and run-time environment

You can set the Hive version, run-time environment, and Hadoop connection with a parameter.

For more information about mapping parameters, see the *Informatica Developer Mapping Guide*.

Parameter Usage

Use parameters for big data sources or target properties, connection properties, and run-time environment properties.

Big Data Sources and Targets

Hive sources

You can configure the following parameters for Hive Read transformation properties:

- Connection. Configure this parameter on the **Run-time** tab.
- Owner. Configure this parameter on the **Run-time** tab.
- Resource. Configure this parameter on the **Run-time** tab.
- Joiner queries. Configure this parameter on the **Query** tab.
- Filter queries. Configure this parameter on the **Query** tab.
- PreSQL commands. Configure this parameter on the **Advanced** tab.
- PostSQL commands. Configure this parameter on the **Advanced** tab.
- Constraints. Configure this parameter on the **Advanced** tab.

HBase sources and targets

You can configure the following parameters for HBase Read and Write transformation properties:

- Connection. Configure this parameter on the **Overview** tab.
- Date Time Format for the Read or Write data object. Configure this parameter on the **Advanced** tab.

Complex file sources and targets

You can configure the following parameters for complex file Read and Write transformation properties:

- Connection. Configure this parameter on the **Overview** tab.
- Data object read operation. Configure the following parameters on the **Advanced** tab:
  - File Path
  - File Format.

- Input Format
- Compression Format
- Custom Compression Codec properties

- Data object write operation. Configure the following parameters on the **Advanced** tab:
  - File Name
  - File Format
  - Output Format
  - Output Key Class
  - Output Value Class
  - Compression Format
  - Custom Compression Codec
  - Sequence File Compression Type

Flat file on HDFS sources and targets

You can configure the following parameters for a flat file on HDFS Read and Write transformation properties:

- Data object read operation. Configure the following parameters on the **Run-time** tab:
  - Source File Name
  - Source File Directory
- Data object write operation. Configure the following parameters on the **Run-time** tab:
  - Output File Directory
  - Output File Name

Hadoop connection and run-time environment

You can configure the following mapping parameters on the **Run-time** tab for a mapping in the Hadoop environment:

- Hive version.
- Run-time environment.
- Hadoop connection.

Create and Use Hadoop Parameters

Create parameters to rerun a mapping with different values in the Hadoop environment. You can use parameters for sources or targets and at the mapping-level.

The following high-level steps describe how to create and use parameters in the Hadoop environment.

- Create parameters for big data sources or targets.
- Create mapping-level parameters.
- Create parameter sets.
- Create an application.
- Optionally, generate a parameter file.
- Run the mapping with a parameter set or optionally, run the mapping with a parameter file.

Workflows that Run Mappings in a Hadoop Environment

You can add a mapping configured to run in a Hadoop 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.

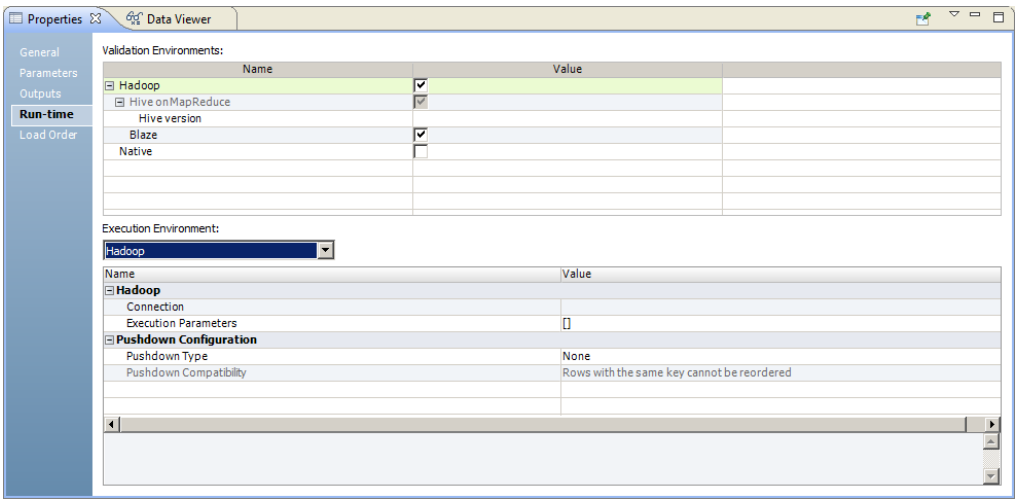
When a Mapping task runs a mapping configured to run in a Hadoop environment, do not assign the Mapping task outputs to workflow variables. Mappings that run in a Hadoop environment do not provide the total number of target, source, and error rows. When a Mapping task includes a mapping that runs in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

Configuring a Mapping to Run in a Hadoop Environment

You can configure a mapping to run in a Hadoop environment. To configure a mapping, you must select the Hadoop environment and a Hadoop connection.

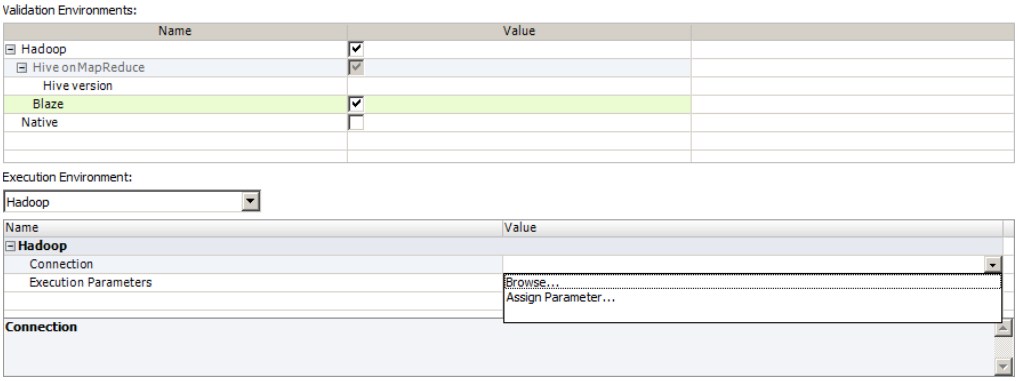
1. Select a mapping from a project or folder from the **Object Explorer** view to open in the editor.
2. In the **Properties** view, select the **Run-time** tab.
3. Select **Hadoop** as the value for the validation environment.

   The **Hive on MapReduce** and **Blaze** engines are selected by default. To only use the **Hive on MapReduce** engine, clear the **Blaze** engine. If you use the **Blaze** engine, you cannot clear the **Hive on MapReduce** engine.
4. In the execution environment, select **Hadoop**.



5. In the Hadoop environment, select **Connection** and use the drop down in the value field to browse for a connection or select a connection parameter:

- To select a connection, click **Browse** and select a connection.
- To select a connection parameter, click **Assign Parameter**.



6. Optionally, select **Execution Parameters** and select a Hadoop connection parameter that you defined on the **Parameters** tab for the mapping.
7. Right-click an empty area in the editor and click **Run Mapping**.

Mapping Execution Plans

When you run a mapping in a Hadoop environment, the Data Integration Service generates a Hive or Blaze engine execution plan for the mapping.

When the Data Integration Service uses the Hive engine, it 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.

When the Data Integration Service uses the Blaze engine, it has a Blaze engine executor that can process the mapping. The Blaze engine executor simplifies the mapping to segments and generates a Blaze engine execution plan. Each segment can contain multiple tasklets. Each tasklet can contain multiple partitions.

The Blaze engine execution plan contains tasks to start the mapping, run the mapping, and clean up the temporary tables and files. It contains the number of tasklets in a mapping, tasklet details, and the task recovery strategy. It also contains pre and post grid task preparation commands for each mapping before running the main mapping on a Hadoop cluster. A pre-grid task can include a task such as copying data to HDFS. A post-grid task can include tasks such as cleaning up temporary files or copying data from HDFS.

Hive Engine Execution Plan Details

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

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

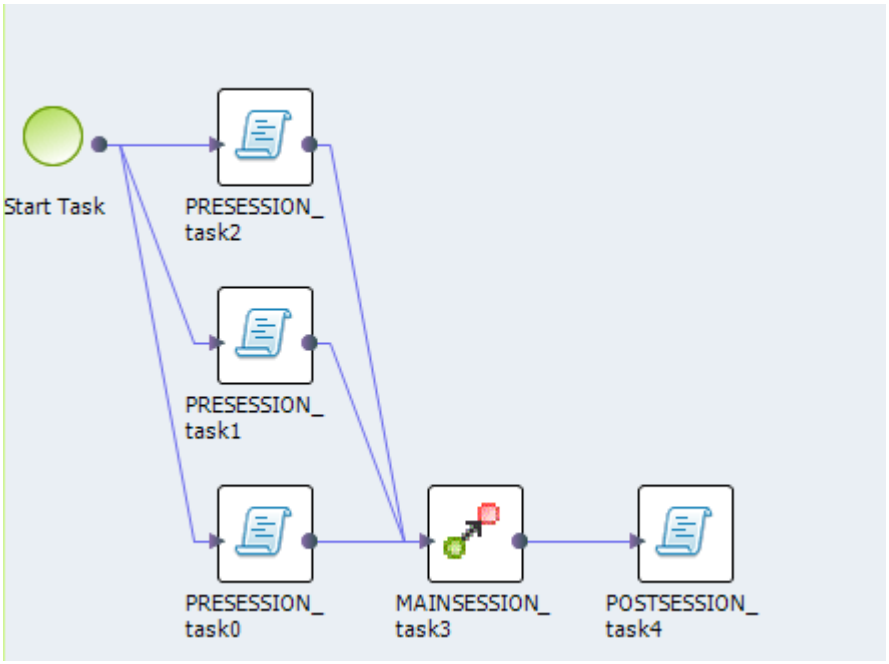
Property	Description
Script Name	Name of the Hive script.
Script	Hive script that the Data Integration Service generates based on the mapping logic.
Depends On	Tasks that the script depends on. Tasks include other scripts and Data Integration Service tasks, like the Start task.

Blaze Engine Execution Plan Details

You can view details of the Blaze engine execution plan in the Administrator tool and Developer tool.

In the Developer tool, the Blaze engine execution plan appears as a workflow. You can click on each component in the workflow to get the details.

The following image shows the Blaze execution plan in the Developer tool:



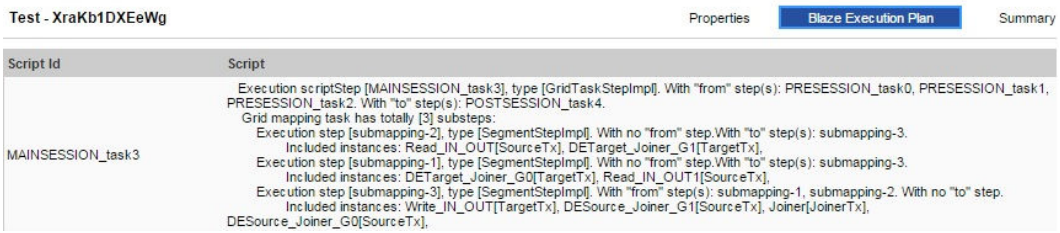
The Blaze engine execution plan workflow contains the following components:

- Start task. The workflow start task.
- Command task. The pre-processing or post-processing task for local data.
- Grid mapping. An Informatica mapping that the Blaze engine compiles and distributes across a cluster of nodes.
- Grid task. A parallel processing job request sent by the Blaze engine executor to the Grid Manager.

- Grid segment. Segment of a grid mapping that is contained in a grid task.
- Tasklet. A partition of a grid segment that runs on a separate DTM.

In the Administrator tool, the Blaze engine execution plan appears as a script.

The following image shows the Blaze execution script:



In the Administrator tool, the Blaze engine execution plan has the following details:

- Script ID. Unique identifier for the Blaze engine script.
- Script. Blaze engine script that the Data Integration Service generates based on the mapping logic.
- Depends on. Tasks that the script depends on. Tasks include other scripts and Data Integration Service tasks, like the Start task.

Viewing the Execution Plan for a Mapping in the Developer Tool

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

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

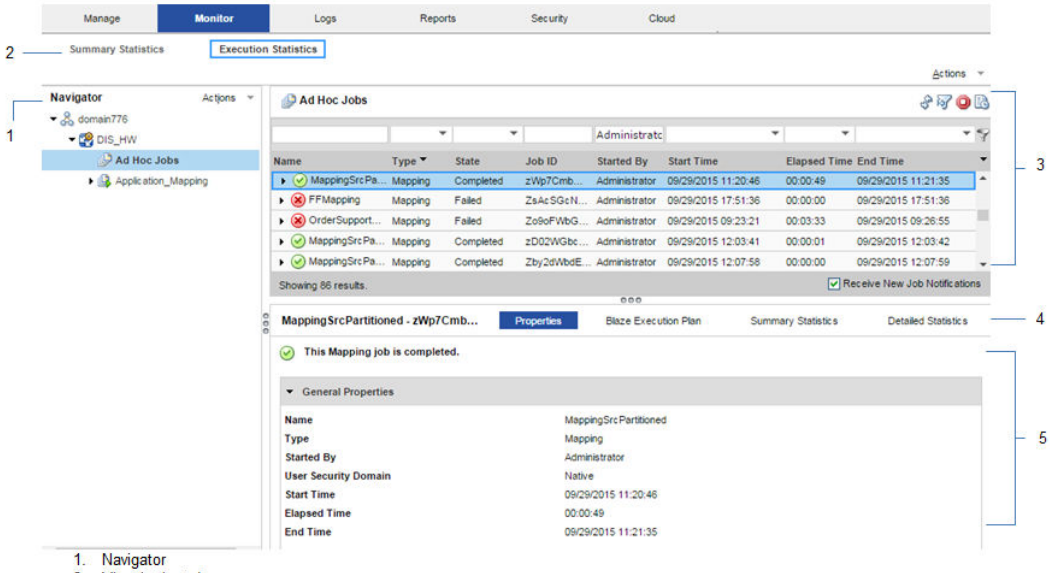
1. To view the execution plan in the Developer tool, select the **Data Viewer** view for the mapping and click **Show Execution Plan**.
2. Select the **Data Viewer** view.
3. Select **Show Execution Plan**.

   The **Data Viewer** view shows the details for the execution plan.

Monitor Jobs

You can monitor statistics and view log events for a mapping job in the Monitoring tab of the Administrator tool. You can also monitor mapping jobs for the Blaze engine in the Blaze Job Monitor web application.

The following image shows the Monitor tab in the Administrator tool:



1. Navigator
2. View in the tab.
3. Contents panel
4. View in the contents panel.
5. Details panel

The Monitor tab has the following views:

Summary Statistics

Use the **Summary Statistics** view to view graphical summaries of object states and distribution across the Data Integration Services. You can also view graphs of the memory and CPU that the Data Integration Services used to run the objects.

Execution Statistics

Use the **Execution Statistics** view to monitor properties, run-time statistics, and run-time reports. In the Navigator, you can expand a Data Integration Service to monitor **Ad Hoc Jobs** or expand an application to monitor deployed mapping jobs.

When you select **Ad Hoc Jobs** or deployed mapping jobs from an application in the Navigator of the **Execution Statistics** view, a list of jobs appears in the contents panel. The contents panel groups related jobs based on the job type. You can expand a job type to view the related jobs under it.

The following views can appear when you view a job in the **Execution Statistics** view:

Properties

The **Properties** view shows the general properties about the selected job such as name, job type, user who started the job, and start time of the job. If you ran the mapping with the Blaze engine, you can also monitor jobs on the Hadoop cluster from the Monitoring URL that appears for the mapping in the general properties. The Monitoring URL opens the Blaze Job Monitor in a web page. The Blaze Job Monitor displays detailed monitoring statistics for a mapping such as the number of grid tasks, grid segments, or tasklets, and recovery attempts for each tasklet.

Blaze Execution Plan

The **Blaze Execution Plan** view appears when you run a mapping with the Blaze engine in the Hadoop environment and displays the execution plan for the Blaze engine mapping. The Blaze execution plan displays the Blaze engine script that the Data Integration Service generates based on the mapping logic, a unique identifier for the script, and the tasks that the script depends on.

Hive Execution Plan

The **Hive Execution Plan** view appears when you run a mapping with the Hive engine in the Hadoop environment and displays the execution plan for the Hive engine mapping. The Hive execution plan displays the Hive script that the Data Integration Service generates based on the mapping logic, a unique identifier for the script, and the tasks that the script depends on.

Summary Statistics

The **Summary Statistics** view appears in the details panel when you select a mapping job in the contents panel. The **Summary Statistics** view displays throughput and resource usage statistics for the job run.

Detailed Statistics

The **Detailed Statistics** view appears in the details panel when you select a mapping job in the contents panel. The **Detailed Statistics** view displays graphs of the throughput and resource usage statistics for the job run.

Accessing the Monitoring URL

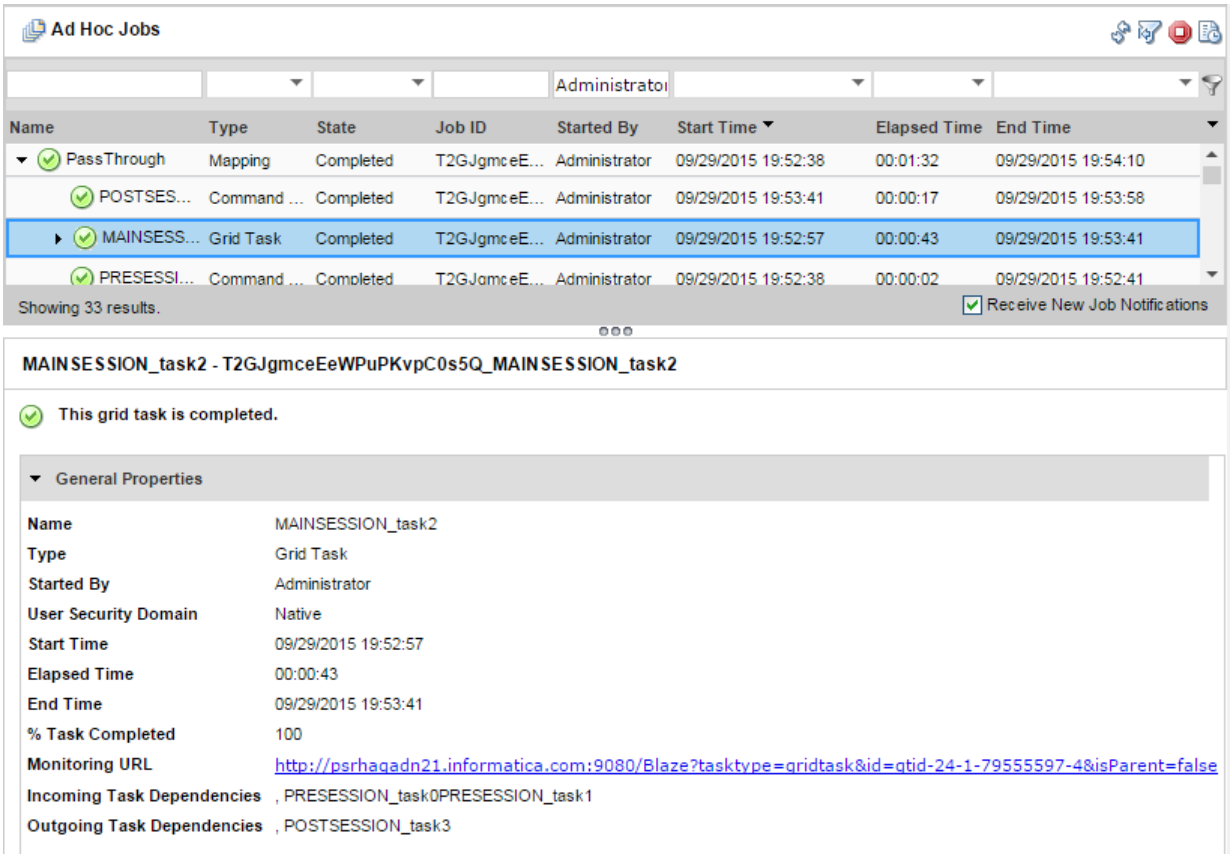
The Monitoring URL for the Blaze engine opens the Blaze Job Monitor web application. You can access the Monitoring URL from the **Execution Statistics** view in the Administrator tool.

1. In the **Monitor** tab of the Administrator tool, click the **Execution Statistics** view.
2. Select **Ad Hoc Jobs** or select a deployed mapping job from an application in the Navigator.

   The list of jobs appears in the contents panel.

3. Select a mapping job and expand the mapping to select a grid task for the mapping.

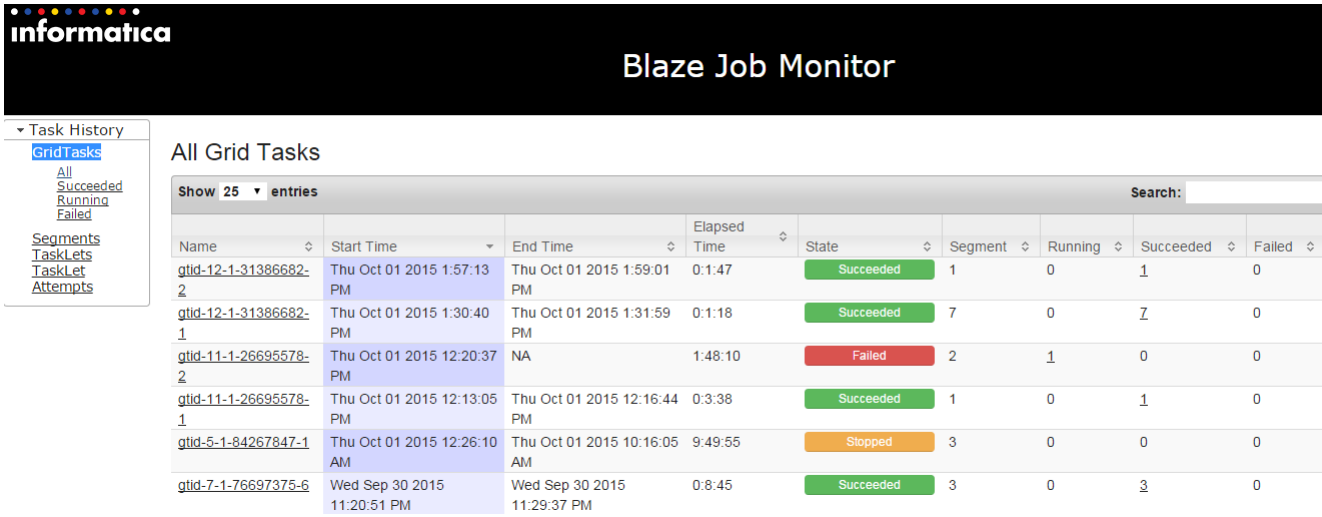
   The Monitoring URL appears in the **Properties** view.



Monitor Blaze Engine Jobs

Use the Blaze Job Monitor application to monitor Blaze engine jobs on the Hadoop cluster. The Blaze engine monitoring URL appears in the Monitor tab of the Administrator tool when you view a Blaze engine mapping job. When you click the URL, the Blaze engine monitoring application opens in a web page.

The following image shows the Blaze Job Monitor:



You can filter Blaze mapping jobs by the following task history :

- Grid task. A parallel processing job request sent by the Blaze engine executor to the Grid Manager. You can view filter by all tasks, or succeeded tasks, running tasks, or failed tasks.
- Grid segment. Part of a grid mapping that is contained in a grid task.
- Tasklet. A partition of a grid segment that runs on a separate DTM.
- Tasklet Attempts. The number of recovery attempts to restart a tasklet.

The Blaze Job Monitor displays the task history for mapping jobs. You can monitor properties for a task such as name, start time, end time, or state of the task. You can also view log events. If you filter mapping jobs by grid segment, you can mouse over a grid segment to view the logical name of the segment.

By default, the Blaze Job Monitor automatically refreshes the list of tasks every five seconds and reverts to the first page that displays tasks. Disable auto refresh if you want to browse through multiple pages. To turn off automatic refresh, click **Action** > **Disable Auto Refresh**.

The Blaze Job Monitor displays the first 100,000 grid tasks run in the past seven days. The Blaze Job Monitor displays the grid segments, tasklets, and tasklet attempts for grid tasks that are running and grid tasks that were accessed in the last 30 minutes.

Monitoring a Mapping

You can monitor a mapping that runs in the Hadoop environment.

1. In the Administrator tool, click the **Monitor** tab.
2. Select the **Execution Statistics** view.
3. In the Navigator, choose to open an ad hoc job or a deployed mapping job.
   - To choose an ad hoc job, expand a Data Integration Service and click **Ad Hoc Jobs**.
   - To choose a deployed mapping job, expand an application and click **Deployed Mapping Jobs**.

   The list of jobs appears in the contents panel.
4. Click a job to view its properties.

   The contents panel shows the default **Properties** view for the job. For a Blaze engine mapping, the Blaze engine monitoring URL appears in the general properties in the details panel.
5. Choose a view in the contents panel to view more information about the job:
   - To view the Blaze engine execution plan for the mapping, select the **Blaze Execution Plan** view.
   - To view the Hive execution plan for the mapping, select the **Hive Execution Plan** view.
   - To view the summary statistics for a job, click the **Summary Statistics** view.
   - To view the detailed statistics for a job, click the **Detailed Statistics** view.

Hadoop Environment Logs

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

You can view logs for the Hive on MapReduce engine or the Blaze engine. You can view log events relating to different types of errors such as Hadoop connection failures, Hive query failures, Hive command failures, or other Hadoop job failures.

You can view reject files for the Hive engine in the reject file directory specified for the Data Integration Service. You cannot view reject files even if you configure a reject file directory for the Blaze engine on the Hadoop cluster nodes.

Blaze Engine Logs

The Blaze engine logs appear in the LDTM log and in the Hadoop cluster logs.

You can find the information about Blaze engine log events in the following log files:

LDTM log

The LDTM logs the results of the Blaze engines execution plan run for the mapping. You can view the LDTM log from the Developer tool or the Monitoring tool for a mapping job.

Blaze component and tasklet logs

The Blaze engine stores tasklet and Blaze component log events in temporary and permanent directories on the Hadoop cluster. The log file directories are specified by properties in the hadoopEnv.properties file located in the following location for each Hadoop distribution:

```
<Informatica Installation directory>/services/shared/hadoop/<distribution directory>/
infaConf
```

The temporary directory is specified by the following property in the hadoopEnv.properties file: `infagrid.node.local.root.log.dir`. An administrator must create a directory with read, write, and execute permissions on all nodes on the Hadoop cluster.

For example, configure the following path for the property:

```
infagrid.node.local.root.log.dir=$HADOOP_NODE_INFA_HOME/dtmLogs
```

After the mapping completes, the Data Integration Service moves the tasklet log events from the temporary directory to a permanent directory on HDFS. The permanent directory is specified by the following property in the hadoopEnv.properties file: `infacal.hadoop.logs.directory`.

For example, configure the following path for the property:

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

If you want to retain the tasklet logs in the temporary directory, set the value of the following property in the hadoopEnv.properties file to false: `infagrid.delete.local.log`

If you do not configure the temporary or permanent directories, the tasklet log events appear in the directory configured for the DTM Process. You can get the directory for the DTM Process from the value for the `yarn.nodemanager.local-dirs` property in yarn-site.xml on the cluster node.

The following sample code describes the `yarn.nodemanager.local-dirs` property:

```
<property>
<name>yarn.nodemanager.local-dirs</name>
<value>/var/lib/hadoop-yarn/cache/${user.name}/nm-local-dir</value>
<description>List of directories to store local files.</description>
</property>
```

Hive Engine Logs

The Hive engine logs appear in the LDTM log and the Hive session log.

You can find the information about Hive engine log events in the following log files:

LDTM log

The LDTM logs the results of the Hive queries run for the mapping. You can view the LDTM 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.

Viewing Hadoop Environment Logs in the Administrator Tool

You can view log events for a Blaze or Hive mapping from the Monitor tab of the Administrator tool.

1. In the Administrator tool, click the **Monitor** tab.
2. Select the **Execution Statistics** view.
3. In the Navigator, choose to open an ad hoc job or a deployed mapping job.
   - To choose an ad hoc job, expand a Data Integration Service and click **Ad Hoc Jobs**.
   - To choose a deployed mapping job, expand an application and click **Deployed Mapping Jobs**.

   The list of jobs appears in the contents panel.
4. Click **Actions** > **View Logs for Selected Object**to view the run-time logs for the mapping.

   The log file shows the results of the Hive queries and Blaze engine queries run by the Data Integration Service. This includes the location of Hive session logs and Hive session history file.

Viewing Logs in the Blaze Job Monitor

You can view logs for a Blaze mapping from the Blaze Job Monitor.

1. In the Blaze Job Monitor, select a job from the list of jobs.
2. In the row for the selected job, click the **Logs** link.

   The log events appear in another browser window.

Optimization for the Hadoop Environment

You can optimize the Hadoop environment and the Hadoop cluster to increase performance.

You can optimize the Hadoop environment and the Hadoop cluster in the following ways:

Configure a highly available Hadoop cluster

You can configure the Data Integration Service and the Developer tool to read from and write to a highly available Hadoop cluster. The steps to configure a highly available Hadoop cluster depend on the type of Hadoop distribution. For more information about configuration steps for a Hadoop distribution, see the *Informatica Big Data Management Installation and Configuration Guide*.

Truncate partitions in a Hive target

You can truncate partitions in a Hive target to increase performance. To truncate partitions in a Hive target, you must choose to both truncate the partition in the Hive target and truncate the target table. You can enable data compression on temporary staging tables to optimize performance.

Compress data on temporary staging tables

You can enable data compression on temporary staging tables to increase mapping performance.

Parallel sort

When you use a Sorter transformation in a mapping, the Data Integration Service enables parallel sorting by default when it pushes the mapping logic to the Hadoop cluster. Parallel sorting improves mapping performance with some restrictions.

Truncating Partitions in a Hive Target

To truncate partitions in a Hive target, you must edit the write properties for the customized data object that you created for the Hive target in the Developer tool.

1. Open the customized data object in the editor.
2. To edit write properties, select the **Input** transformation in the **Write** view, and then select the **Advanced** properties.
3. Select **Truncate Hive Target Partition**.
4. Select **Truncate target table**.

Enabling Data Compression on Temporary Staging Tables

To optimize performance when you run a mapping in the Hadoop environment, you can enable data compression on temporary staging tables. When you enable data compression on temporary staging tables, mapping performance might increase.

To enable data compression on temporary staging tables, complete the following steps:

1. Configure the Hive connection to use the codec class name that the Hadoop cluster uses to enable compression on temporary staging tables.
2. Configure the Hadoop cluster to enable compression on temporary staging tables.

Hadoop provides following compression libraries for the following compression codec class names:

Compression Library	Codec Class Name	Performance Recommendation
Zlib	org.apache.hadoop.io.compress.DefaultCodec	n/a
Gzip	org.apache.hadoop.io.compress.GzipCodec	n/a
Snappy	org.apache.hadoop.io.compress.SnappyCodec	Recommended for best performance.
Bz2	org.apache.hadoop.io.compress.BZip2Codec	Not recommended. Degrades performance.
LZO	com.hadoop.compression.lzo.LzoCodec	n/a

Step 1. Configure the Hive Connection to Enable Data Compression on Temporary Staging Tables

Use the Administrator tool or the Developer tool to configure the Hive connection. You can edit the Hive connection properties to configure the codec class name that enables data compression on temporary staging tables.

1. In the Hive connection properties, edit the properties to run mappings in a Hadoop cluster.
2. Select **Temporary Table Compression Codec**.
3. Choose to select a predefined codec class name or enter a custom codec class name.

- To select a predefined codec class name, select a compression library from the list.
- To enter a custom codec class name, select custom from the list and enter the codec class name that matches the codec class name in the Hadoop cluster.

Step 2. Configure the Hadoop Cluster to Enable Compression on Temporary Staging Tables

To enable compression on temporary staging tables, you must install a compression codec on the Hadoop cluster.

For more information about how to install a compression codec, refer to the Apache Hadoop or Hive documentation.

1. Verify that the native libraries for the compression codec class name are installed on every node on the cluster.
2. To include the compression codec class name that you want to use, update the property io.compression.codecs in core-site.xml. The value for this property is a comma separated list of all the codec class names supported on the cluster.
3. Verify that the Hadoop-native libraries for the compression codec class name that you want to use are installed on every node on the cluster.
4. Verify that the LD_LIBRARY_PATH variable on the Hadoop cluster includes the locations of both the native and Hadoop-native libraries where you installed the compression codec.

Parallel Sorting

To improve mapping performance, the Data Integration Service enables parallel sorting by default in a mapping that has a Sorter transformation and a flat file target.

The Data Integration Service enables parallel sorting for mappings in a Hadoop environment based on the following rules and guidelines:

- The mapping does not include another transformation between the Sorter transformation and the target.
- The data type of the sort keys does not change between the Sorter transformation and the target.
- Each sort key in the Sorter transformation must be linked to a column in the target.

Troubleshooting a Mapping in a Hadoop 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 Hadoop environment, the Hive connection selected for the Hive source or Hive target, and the mapping must be on the same Hive metastore.

When I run a mapping with MapR 2.1.2 distribution that processes large amounts of data, monitoring the mapping from the Administrator tool stops.

You can check the Hadoop task tracker log to see if there a timeout that results in Hadoop job tracker and Hadoop task tracker losing connection. To continuously monitor the mapping from the Administrator tool, increase the virtual memory to 640 MB in the hadoopEnv.properties file. The default

is 512 MB. For example, `infapdo.java.opts=-Xmx640M -XX:GCTimeRatio=34 -XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:ParallelGCThreads=2 -XX:NewRatio=2 -Djava.library.path=$HADOOP_NODE_INFA_HOME/services/shared/bin:$HADOOP_NODE_HADOOP_DIST/lib/native/Linux-amd64-64 -Djava.security.egd=file:/dev/./urandom -Dmapr.library.flatclass`

When I run a mapping with a Hadoop distribution on MapReduce 2, the Administrator tool shows the percentage of completed reduce tasks as 0% instead of 100%.

Verify that the Hadoop jobs have reduce tasks.

When the Hadoop distribution is on MapReduce 2 and the Hadoop jobs do not contain reducer tasks, the Administrator tool shows the percentage of completed reduce tasks as 0%.

When the Hadoop distribution is on MapReduce 2 and the Hadoop jobs contain reducer tasks, the Administrator tool shows the percentage of completed reduce tasks as 100%.

CHAPTER 4

Mappings in the Native Environment

This chapter includes the following topics:

Mappings in the Native Environment Overview

You can run a mapping in the native 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

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

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 Hadoop 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 data type.
   - 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 Hadoop 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 Hadoop 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
- PowerExchange for Twitter
- PowerExchange for Web Content-Kapow Katalyst

Review the respective PowerExchange adapter documentation for more information.

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.

CHAPTER 5

Profiles

This chapter includes the following topics:

Profiles Overview

You can run a profile on HDFS and Hive data sources in the Hadoop environment when you use the Hive engine. 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 or Informatica Analyst, you can choose either native or Hadoop run-time environment to run a profile. If you choose the Hadoop environment, Informatica Developer or Informatica Analyst sets the run-time environment in the profile definition.

When you run a profile on in the Hadoop environment from the Developer tool, you validate the data source before you run the profile. To validate the data source, 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 and 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.

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, patterns, and data types. You can run a data domain profile in the Hadoop environment to discover source column data that match predefined data domains. The predefined data domains might contain column data rules or column name rules. You can also run a profile that has associated rules in the Hadoop environment.

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

In the Analyst tool, you can edit a profile that you ran on an HDFS source in the Developer tool. In the Analyst tool, you can create a scorecard based on a profile that you ran on an HDFS source in the Developer tool. Scorecards run in the native environment.

Column Profiles on Hadoop

You can import a native, Hive, and HDFS data source into the Analyst tool or 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 data types.

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

1. Open a connection in the Analyst tool or Developer tool to import the native or Hadoop source.
2. Import the data source as a data object. The Analyst tool or 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 the run-time settings and 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.

Data Domain Discovery on Hadoop

Data domain discovery is the process of discovering column names or column data in the data sources that match predefined formats. You can run a data domain profile on Hadoop and view the results.

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.

Running a Profile on Hadoop in the Developer Tool

After you set up the validation and run-time environments for a profile in the Developer tool, 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 profile options and data domain discovery options.

7. Click **Run Settings**.

   The **Run Settings** pane appears.
8. Select **Hadoop** as the validation environment.

   You can select both **Native** and **Hadoop** as the validation environments.
9. Select **Hadoop** as the run-time environment.
10. Select a Hive connection.
11. Click **Finish**.

Running a Profile on Hadoop in the Analyst Tool

When you create or edit a profile in the Analyst tool, you can select the run-time environment.

1. In the **Discovery Home** panel, click **Data Object Profile** or select **New** > **Data Object profile** from anywhere in the Analyst tool.

   The **New Profile** wizard appears. The **Column profiling** option is selected by default.
2. Click **Next**.
3. In the **Sources** pane, select a data object.
4. Click **Next**.
5. Enter a name and an optional description for the profile.
6. In the **Folders** pane, select the project or folder where you want to create the profile.

   The Analyst tool displays the project that you selected and shared projects that contain folders where you can create the profile. The profiles in the folder appear in the right pane.
7. Click **Next**.
8. In the **Columns** pane, select the columns that you want to run a profile on. The columns include any rules that you applied to the profile. The Analyst tool lists column properties, such as the name, data type, precision, and scale for each column.

   Optionally, select **Name** to select all columns.
9. In the **Sampling Options** pane, configure the sampling options.
10. In the **Drilldown Options** pane, configure the drill-down options.

    Optionally, click **Select Columns** to select columns to drill down on. In the **Drilldown columns** dialog box, select the columns for drilldown and click **OK**.
11. Accept the default option in the **Profile Results Option** pane.

    The first time you run the profile, the Analyst tool displays profile results for all columns selected for profiling.
12. Click **Next**.
13. Optionally, define a filter for the profile.
14. Click **Next** to verify the row drill-down settings including the preview columns for drilldown.
15. To run the profile in the Hadoop environment, select **Hive** and then select a hive connection. The Hive connection helps the Data Integration Service communicate with the Hadoop cluster to push down the profile execution from the Data Integration Service to the Hadoop cluster.
16. Click **Save** to create the profile, or click **Save & Run** to create the profile and then run the profile.

Running Multiple Data Object Profiles on Hadoop

You can run a column profile on multiple data source objects in the Developer tool. 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 Hadoop settings.
8. Select **Hadoop** and select a Hive connection.

   You can select both **Native** and **Hadoop** as the validation environments.
9. In the **Run-time Environment** field, select **Hadoop**.
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.

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 Hadoop 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 occured because of an issue with the Hadoop cluster. Contact your network administrator to resolve the issue.

CHAPTER 6

Native Environment Optimization

This chapter includes the following topics:

Native Environment Optimization Overview

You can optimize the native environment to increase performance. To increase performance, you can configure the Data Integration Service to run on a grid and to use multiple partitions to process data. You can also enable high availability to ensure that the domain can continue running despite temporary network, hardware, or service failures.

You can run profiles, sessions, and workflows on a grid to increase the processing bandwidth. A grid is an alias assigned to a group of nodes that run profiles, sessions, and workflows. When you enable grid, the Data Integration Service runs a service process on each available node of the grid to increase performance and scalability.

You can also run mapping with partitioning to increase performance. When you run a partitioned session or 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 a Grid

You can run an Integration Service on a grid to increase the processing bandwidth. When you enable grid, the Integration Service runs a service process on each available node of the grid to increase performance and scalability.

Big data may require additional bandwidth to process large amounts of data. For example, when you run a Model repository profile on an extremely large data set, the Data Integration Service grid splits the profile into multiple mappings and runs the mappings simultaneously on different nodes in the grid.

Data Integration Service Grid

You can run Model repository mappings and profiles on a Data Integration Service grid.

When you run mappings on a grid, the Data Integration Service distributes the mappings to multiple DTM processes on nodes in the grid. When you run a profile on a grid, the Data Integration Service splits the profile into multiple mappings and distributes the mappings to multiple DTM processes on nodes in the grid.

For more information about the Data Integration Service grid, see the *Informatica Administrator Guide*.

Grid Optimization

You can optimize the grid to increase performance and scalability of the Data Integration Service.

To optimize the grid, complete the following task:

Add nodes to the grid.

Add nodes to the grid to increase processing bandwidth of the Data Integration Service.

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 session or mapping which can increase performance.

Partitioned Model Repository Mappings

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

If the nodes where mappings run have multiple CPUs, you can enable the Data Integration Service 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, developers can set a maximum parallelism value for a mapping in the Developer tool. By default, the maximum parallelism for each mapping is set to Auto. Each mapping uses the maximum parallelism value defined for the Data Integration Service. Developers can change the maximum parallelism value in the mapping run-time properties to define a maximum value for a particular mapping. When maximum parallelism is set to different integer values for the Data Integration Service and the mapping, the Data Integration Service uses the minimum value.

For more information, see the *Informatica Application Services Guide* and the *Informatica Developer 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 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 Data Integration Service and the Data 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, consider the differences between internal Informatica components and systems that are external to Informatica. Internal components include the

Service Manager, application services, 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.

APPENDIX A

Data Type Reference

This appendix includes the following topics:

Data Type Reference Overview

Informatica Developer uses the following data types in Hive mappings:

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

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

Hive Complex Data Types

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

Note: Hive complex data types 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 data types:

Complex Data Type	Description
Array	The elements in the array are of string data type. The elements in the array are delimited by commas. For example, an array of `fruits` is represented as `[apple,banana,orange]`.
Map	Maps contain key-value pairs and are represented as pairs of strings and integers delimited by the = character. String and integer pairs are delimited by commas. For example, a map of `fruits` is represented as `[1=apple,2=banana,3=orange]`.
Struct	Structs are represented as pairs of strings and integers delimited by the : character. String and integer pairs are delimited by commas. For example, a struct of `fruits` is represented as `[1,apple]`.

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:

Hive Data Type	Transformation Data Type	Range and Description
Binary	Binary	1 to 104,857,600 bytes. You can read and write data of Binary data type in a Hadoop environment. You can use the user-defined functions to transform the binary data type.
Tiny Int	Integer	-32,768 to 32,767
Integer	Integer	-2,147,483,648 to 2,147,483,647 Precision 10, scale 0
Bigint	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0

Hive Data Type	Transformation Data Type	Range and Description
Decimal	Decimal	Precision 1 to 28, scale 0 to 28 For transformations that support precision up to 38 digits, the precision is 1 to 38 digits, and the scale is 0 to 38. For transformations that support precision up to 28 digits, the precision is 1 to 28 digits, and the scale is 0 to 28. For transformations that support precision up to 38 digits, the precision is 1 to 38 digits, and the scale is 0 to 38. For transformations that support precision up to 28 digits, the precision is 1 to 28 digits, and the scale is 0 to 28. If a mapping is not enabled for high precision, the Data Integration Service converts all decimal values to double values. If a mapping is enabled for high precision, the Data Integration Service converts decimal values with precision greater than 38 digits to double values.
Double	Double	Precision 15
Float	Double	Precision 15
String	String	1 to 104,857,600 characters
Boolean	Integer	1 or 0 The default transformation type for boolean is integer. You can also set this to string data type with values of True and False.
Arrays	String	1 to 104,857,600 characters
Struct	String	1 to 104,857,600 characters
Maps	String	1 to 104,857,600 characters
Timestamp	datetime	The time stamp format is YYYY-MM-DD HH:MM:SS.fffffffff. Precision 29, scale 9.
Date	datetime	0000-0101 to 999912-31. Hive date format is YYYY-MM-DD. Precision 10, scale 0.

INDEX

M

N

O

P

R

S

V

W