Informatica

Informatica® B2B Data Transformation
10.1.1

HParser Operator Guide

Informatica B2B Data Transformation HParser Operator Guide
10.1.1
December 2016

© Copyright Informatica LLC 2012, 2018

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. 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.

Informatica, and the Informatica logo are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at https://www.informatica.com/trademarks.html. Other company and product names may be trade names or trademarks of their respective owners.

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; and http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/ intro.html.

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.

This Software is protected by U.S. Patent Numbers 5,794,246; 6,014,670; 6,016,501; 6,029,178; 6,032,158; 6,035,307; 6,044,374; 6,092,086; 6,208,990; 6,339,775; 6,640,226; 6,789,096; 6,823,373; 6,850,947; 6,895,471; 7,117,215; 7,162,643; 7,243,110; 7,254,590; 7,281,001; 7,421,458; 7,496,588; 7,523,121; 7,584,422; 7,676,516; 7,720,842; 7,721,270; 7,774,791; 8,065,266; 8,150,803; 8,166,048; 8,166,071; 8,200,622; 8,224,873; 8,271,477; 8,327,419; 8,386,435; 8,392,460; 8,453,159; 8,458,230; 8,707,336; 8,886,617 and RE44,478, International Patents and other Patents Pending.

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

NOTICES

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

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

The information in this documentation is subject to change without notice. If you find any problems in this documentation, please report them to us in writing at Informatica LLC 2100 Seaport Blvd. Redwood City, CA 94063.

INFORMATICA LLC PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2018-06-26

Table of Contents

Preface 6

Informatica Resources. 6

Informatica My Support Portal. 6

Informatica Documentation. 6

Informatica Product Availability Matrixes. 6

Informatica Web Site. 7

Informatica How-To Library. 7

Informatica Knowledge Base. 7

Informatica Support YouTube Channel. 7

Informatica Marketplace. 7

Informatica Velocity. 7

Informatica Global Customer Support. 7

Chapter 1: Introduction to HParser. 9

HParser Overview. 9

Data Transformation HParser Architecture. 9

Hadoop MapReduce. 11

Command Node. 11

Chapter 2: Data Transformation HParser Installation. 12

HParser Installation Overview. 12

HParser Installation Prerequisites. 12

Installing Data Transformation HParser. 13

Changing Permissions on the Directories. 13

Configuring the Command Node. 14

Installing the Informatica Services and Client with Data Transformation. 14

Uninstalling Data Transformation HParser. 15

Chapter 3: Running Data Transformation on Hadoop. 16

Overview. 16

Running a Data Transformation Service on Hadoop 16

Create a Data Processor Transformation. 17

Data Processor Transformation Development. 17

Create the Data Processor Transformation. 17

Select the Schema Objects 18

Create Objects in a Blank Data Processor Transformation. 18

Creating a Script. 18

Creating an XMap. 19

Create the Ports. 19

Deploy the Data Transformation Service to the Data Transformation Repository. 20

Create a Directory for the Input and Output Files 20
Run the MapReduce Job. 20
MapReduce Command Syntax. 20
Input Parameters (-D). 21
Define the Input Format. 23
Mapper Options. 24
Reducer Options. 25
Output Options. 25
Troubleshooting. 25
HParser Error Messages. 26
Task Operation Messages. 26

Chapter 4: Example. 28

Example Overview. 28
Step 1. Copy the Data Transformation Service to the Data Transformation Repository. 28
Step 2. Create a Directory for the Input and Output Files. 29
Step 3. Run the MapReduce Job. 29

Index. 30

Preface

The *Data Transformation HParser Operator Guide* describes how to install and use Data Transformation HParser. It is written for system administrators who need to run a Data Transformation service as a distributed application. This guide assumes that the reader understands Hadoop, the HDFS file system, and Data Transformation.

Informatica Resources

Informatica My Support Portal

As an Informatica customer, you can access the Informatica My Support Portal at http://mysupport.informatica.com.

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

The site contains product information, user group information, newsletters, access to the Informatica How-To Library, the Informatica Knowledge Base, Informatica Product Documentation, and access to the Informatica user community.

Informatica Documentation

The Informatica Documentation team 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 http://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/community/my-support/product-availability-matrices.

Informatica Web Site

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

Informatica How-To Library

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

Informatica Knowledge Base

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

Informatica Support YouTube Channel

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

Informatica Marketplace

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

Informatica Velocity

You can access Informatica Velocity at http://mysupport.informatica.com. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Global Customer Support

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

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

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 HParser

This chapter includes the following topics:

- HParser Overview, 9
- Data Transformation HParser Architecture, 9

HParser Overview

HParser is a tool that optimizes Data Transformation services for Hadoop. You can use HParser to define transformations for complex data into flattened, usable formats for Hive, PIG, and MapReduce processing. Hadoop developers can run Data Transformation services as distributed applications on Hadoop.

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.

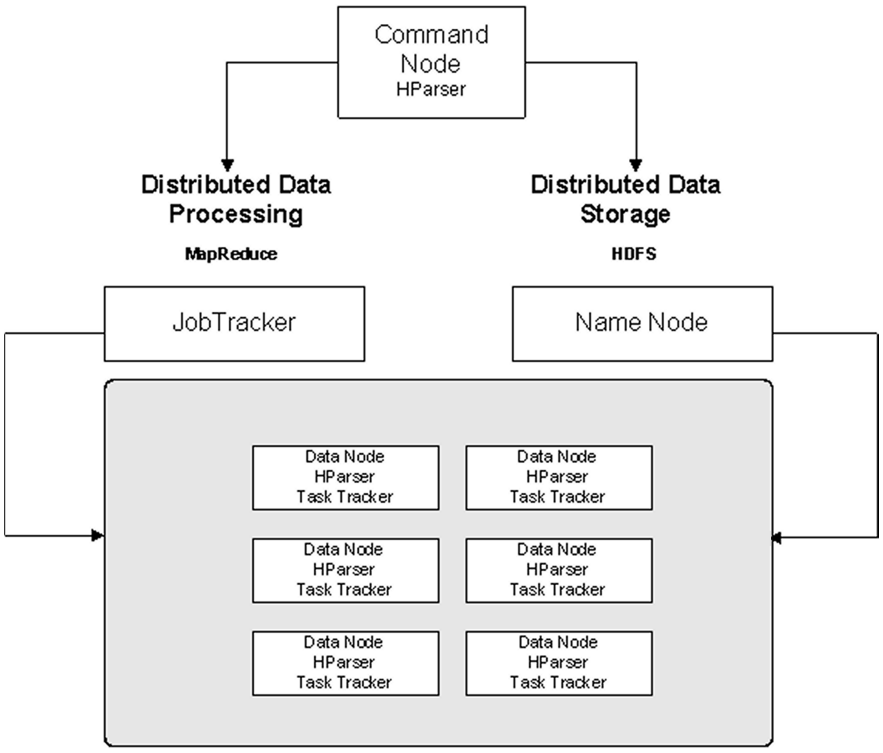
HParser can process large amounts of input data. The Hadoop Data File System (HDFS) splits up the files. You do not have to define how to split, stream, or aggregate the source files.

For example, an application produces hundreds of data files per second and it writes the files to a directory. Data Transformation HParser can assign each file as input to an available instance of a Data Transformation service on Hadoop.

Data Transformation HParser Architecture

HParser runs services on a cluster of nodes. Each node hosts the Hadoop MapReduce framework and an instance of HParser.

The following image shows the HParser on Hadoop architecture:



The HParser architecture contains the following components:

Command Node

A node in the Hadoop cluster from which you run the MapReduce jobs. The command node hosts the Data Transformation HParser JAR file. The JAR file defines the structure of the MapReduce jobs. The JAR file contains methods to run the Data Transformation Engine through the Data Transformation API.

Hadoop Distributed File System (HDFS)

Stores the input and output files for HParser.

Name Node

Coordinates the data storage function on HDFS.

MapReduce

Job that analyzes data and determines how to distribute the processing to the Hadoop cluster.

JobTracker

Service within Hadoop that distributes MapReduce tasks to nodes in the cluster.

TaskTracker

A daemon that accepts tasks from the JobTracker, monitors the processes on the node, and notifies the JobTracker when the tasks complete.

HParser also uses the following components:

Informatica Developer

Use the Developer tool to create Data Processor transformations that define parsing for complex data files. Data Processor transformations contain document pre-processors to extract data from binary formats, including packed data, PDF files, Microsoft Word documents, Web logs, proprietary logs, and

Microsoft Excel documents. You can parse hierarchical data, JSON and XML. The Developer tool runs on Microsoft Windows. It does not run on the Hadoop cluster.

Data Transformation Engine

The processor that runs the services that you define in the Developer tool. The Data Transformation Engine can process input files of any size by streaming and splitting the input. The Data Transformation Engine integrates with MapReduce tasks to transform different data formats through a single data abstraction layer.

Hadoop MapReduce

MapReduce simplifies data processing across large data sets by processing data in parallel across distributed nodes. MapReduce calls the Data Transformation Engine to process the data in parallel.

A MapReduce job contains one or more tasks. For each task within the job, Data Transformation HParser passes input to a Data Transformation service in the cluster. The Data Transformation service transforms the data and returns output to Hadoop. Hadoop writes each output record to a file buffer and separates each record with a record delimiter.

MapReduce performs the following steps to process data:

Map step

The command node splits the input data into smaller pieces to process and it distributes the processing to worker nodes. A worker node might split the data again. MapReduce might create a multi-level distribution of the processing. A worker node processes the smaller problem and passes the results back to its parent node.

Reduce step

The command node receives the results from the worker nodes and combines the results in order to return the output.

The distribution of MapReduce tasks is based on the type of Data Transformation service.

With a parser service, each task in the job processes a file. MapReduce closes the output file when all files in the input path are processed.

With a streamer service, HParser divides the input files into chunks. You can define the size of the chunks. Each task processes the chunks of a single input file and MapReduce closes the output file when Data Transformation processes the last chunk.

Command Node

The command node is a node in the Hadoop cluster that controls the MapReduce jobs. Enter HParser commands on the command node.

The command node contains the Data Transformation HParser JAR file. The JAR file defines the structure of the MapReduce jobs and runs the Data Transformation Engine through the Data Transformation Java API.

CHAPTER 2

Data Transformation HParser Installation

This chapter includes the following topics:

- HParser Installation Overview, 12
- HParser Installation Prerequisites, 12
- Installing Data Transformation HParser, 13
- Changing Permissions on the Directories, 13
- Configuring the Command Node, 14
- Installing the Informatica Services and Client with Data Transformation, 14
- Uninstalling Data Transformation HParser, 15

HParser Installation Overview

Install Data Transformation HParser on each node in a Hadoop cluster.

Complete the following tasks to install HParser:

1. Install HParser with the RPM Package Manager on each node in a Hadoop cluster.

   Note: If you installed Big Data Edition on the Hadoop cluster, do not perform this step.
2. Change the permissions to public on the ServiceDB and log directories.
3. Configure a node in the cluster as the command node.

HParser runs Data Transformation services in the Hadoop environment on Unix machines. However, you use the Developer tool in a Windows environment to create the Data Transformation services that HParser runs. You must install the Informatica services and client with Data Transformation on a separate Windows machine to access the Developer tool.

HParser Installation Prerequisites

Before you install HParser, verify the following prerequisites:

- Linux 64 hardware is configured as a cluster of nodes in the network.

- Each node has 200 MB of disk space.
- Each node has Red Hat Enterprise Linux 5, update 3.
- An HDFS is installed and configured on the cluster.
- Hadoop MapReduce framework is installed and configured on all nodes.
- You must have user root privileges.

Installing Data Transformation HParser

Run the RPM Package Manager to install HParser on each node in the cluster.

Note: If you installed Big Data Edition on the Hadoop cluster, do not install the HParser RPM.

1. Copy the `hparser-engine-<version>.x86_64.rpm` installation file to a directory on the local machine, where <version> is the Data Transformation major version.
2. On the local machine, log in as user **root**.
3. Change to the directory where you copied the installation file.
4. Run the following command to complete the installation, where <version> is the Data Transformation major version:

   ```
   rpm -ivh ./hparser-engine-<version>.x86_64.rpm
   ```

The following table describes the HParser directories after installation is complete:

Directory	Description
/usr/lib/hparser	HParser root directory.
/var/lib/hparser	Contains HParser services.
/var/log/hparser	Contains HParser reports and logs.

Changing Permissions on the Directories

Change the permissions to public on the ServiceDB and log directories.

The HParser installation creates the serviceDB and log folder under the root user. Other users, such as the mapred user, must have permission to write to these folders.

Use the following command to change the permissions:

```
chmod -R 777 /var/lib/hparser/ServicesDB /var/log/hparser
```

Configuring the Command Node

Configure one node in the cluster as the command node for submitting HParser jobs.

Copy the appropriate HParser JAR file from the HParser package to the command node. You can copy the JAR file to any directory that you can access in order to run MapReduce jobs.

The HParser package contains the following JAR files:

hparser-1.1.jar

Use with Apache Hadoop 0.20.205 and previous versions. (CDH3; EMR; MapR)

hparser-1.1-hadoop-0.21

Use with Apache Hadoop 0.21 and later. (CDH4)

Installing the Informatica Services and Client with Data Transformation

The installation of the Informatica services and Informatica client with Data Transformation consists of multiple phases. Install the Informatica services and client on a Windows machine.

1. Before you install the Informatica services, perform the following tasks to plan and prepare for the services installation:
   a. Plan the Informatica domain. Consider the number of nodes in the domain, the application services that will run on each node, the system requirements, and the type of user authentication that the domain will use.
   b. Prepare the databases for the domain. Verify the database requirements and set up the databases.
   c. Set up the machines to meet the Windows requirements to ensure that you can successfully install and run the Informatica services
2. Install the Informatica services.

   Use the server installer to install the Informatica services on a Windows machine. The first time you run the installer, you must create the domain. During the installation on the additional machines, you create worker nodes that you join to the domain.
3. After you install the Informatica services, perform the following tasks to complete the services installation:
   a. Complete the domain configuration. Verify code page compatibility, complete tasks required by the type of user authentication used by the domain, and configure environment variables. Optionally, configure secure communication for the domain.
   b. Prepare to create the application services. Verify operating system requirements for application services and create the users and connections required by the application services.
   c. Create the application services in the required order.
4. Install the Informatica client for Data Transformation. Perform the following steps:
   1. Before you install the client, verify the installation and third-party software requirements for the client.
   2. Use the client installer to install the client on a Windows machine.

3. After you install the client, optionally install additional languages and configure the required environment variables for the client.

For more information about how to install the Informatica services and Informatica client with Data Transformation, see the *Informatica Installation and Configuration Guide*.

Uninstalling Data Transformation HParser

Run RPM Package Manager to uninstall HParser.

▶ Run the following command, where <version> is the Data Transformation major version:

```
rpm -e hparser-engine-<version>
```

CHAPTER 3

Running Data Transformation on Hadoop

This chapter includes the following topics:

- Overview, 16
- Running a Data Transformation Service on Hadoop , 16
- Create a Data Processor Transformation, 17
- Deploy the Data Transformation Service to the Data Transformation Repository, 20
- Create a Directory for the Input and Output Files , 20
- Run the MapReduce Job, 20
- Troubleshooting, 25

Overview

HParser is a Data Transformation Engine that runs Data Processor services in the Hadoop environment on big data. HParser integrates the Data Transformation services into Hadoop MapReduce programming.

A Data Transformation service can be used in the different MapReduce phases, such as InputFormat, Map, Reduce, and OutputFormat. You can create one or more Data Transformation services to serve as HParser phases. For example, you can create a transformation Script that serves as an HParser Mapper, for the Map operation of MapReduce, and a transformation Script that serves as an HParser Reducer, for the Reducer operation of MapReduce.

To implement a MapReduce operation with HParser, in the Developer tool you create Data Processor transformations using the New Transformation wizard. Configure the Data Processor transformations, and then export each as a service. Copy the services to the HParser command node, then use the command line to run HParser for each Data Transformation Service.

Running a Data Transformation Service on Hadoop

To run a Data Transformation service on Hadoop, use the following steps:

1. Create a Data Processor transformation in the Developer tool.

2. Export the transformation as a Data Transformation service and copy the service to the HParser command node.
3. Determine the directories for the input and output files.
4. Configure a Hadoop command to submit a MapReduce job and run the service.

Create a Data Processor Transformation

Create a Data Processor transformation in the Developer tool.

To create a sample Data Processor transformation, you can follow the instructions in the *Data Transformation Getting Started Guide*.

Data Processor Transformation Development

Use the New Transformation wizard to auto-generate a Data Processor transformation, or create a blank Data Processor transformation and configure it later. If you create a blank Data Processor transformation, you must select to create a Script, XMap, Library, or Validation Rules object in the transformation. A Script can parse source documents to hierarchical format, convert hierarchical format to other file formats, or map a hierarchical document to another hierarchical format. An XMap converts an input hierarchical file into an output hierarchical file of another structure. A Library converts an industry messaging type into an XML document with a hierarchy structure or from XML to an industry standard format. Choose the schemas that define the input or output hierarchies.

1. Create the transformation in the Developer tool.
2. For a blank Data Processor transformation, perform the following additional steps:
   a. Add schema references that define the input or output XML hierarchies.
   b. Create a Script, XMap, Library, or Validation Rules object.
3. Configure the input and output ports.
4. Test the transformation.

Create the Data Processor Transformation

Create a Data Processor transformation in the Developer tool. If you create a blank Data Processor transformation, you must then create a Script, XMap, Library, or Validation Rules object in the transformation. Alternatively, you can use the New Transformation wizard to auto-generate a Data Processor transformation.

1. In the Developer tool, click **File** > **New** > **Transformation**.
2. Select the Data Processor transformation and click **Next**.
3. Enter a name for the transformation and browse for a Model repository location to put the transformation.
4. Select whether to create Data Processor transformation with a wizard or to create a blank Data Processor transformation.
5. If you selected to create a blank Data Processor transformation, click **Finish**.

   The Developer tool creates the empty transformation in the repository. The **Overview** view appears in the Developer tool.

6. If you selected to create an Data Processor transformation with a wizard, perform the following steps:
   a. Click **Next**.
   b. Select an input format.
   c. Browse to select a schema, copybook, example file, or specification file if required for certain input formats such as COBOL, JSON, or ASN.1.
   d. Select an output format.
   e. Browse to select a schema, copybook, example file, or specification file if required for the output format.
   f. Click **Finish**. The wizard creates the transformation in the repository.

   The transformation might contain a Parser, Serializer, Mapper, or an object with common components. If you selected a schema, copybook, example file, or specification file the wizard also creates a schema in the repository that is equivalent to the hierarchy in the file.

Select the Schema Objects

Choose the schema objects that define the input or output hierarchies for each XMap or Script component that you plan to create.

You can add schema references on the References view or you can add the schema references when you create Script or XMap objects. A schema object must exist in the Model repository before you can reference it in a Script or XMap.

1. In the Data Processor transformation **Refererences** view, click **Add.**
2. If the schema object exists in the Model repository, browse for and select the schema.
3. If the schema does not exist in the Model repository, click **Create a new schema object** and import a schema object from an .xsd file.
4. Click Finish to add the schema reference to the Data Processor transformation.

Create Objects in a Blank Data Processor Transformation

Create a Script, Library, XMap, or Validation Rules object on the Data Processor transformation **Objects** view. After you create the object, you can open the object from the **Objects** view in order to configure it.

Creating a Script

Create a Script object and define the type of Script component to create. Optionally, you can define a schema reference and example source file.

1. In the Data Processor transformation **Objects** view, click **New.**
2. Enter a name for the Script and click **Next.**
3. Choose to create a Parser or Serializer. Select Other to create a Mapper, Transformer, or Streamer component.
4. Enter a name for the component.
5. If the component is the first component to process data in the transformation, enable **Set as startup component**.
6. Click **Next** if you want to enter a schema reference for this Script. Click **Finish** if you do not want to enter the schema reference.

7. If you choose to create a schema reference, select **Add reference to a Schema Object** and browse for the Schema object in the Model repository. Click **Create a new schema object** to create the Schema object in the Model repository.
8. Click **Next** to enter an example source reference or to enter example text. Click **Finish** if you do not want to define an example source.

   Use a example source to define sample data and to test the Script.
9. If you choose to select an example source, select **File** and browse for the sample file.

   You can also enter sample text in the **Text** area. The Developer tool uses the text to test a Script.
10. Click **Finish.**

    The **Script** view appears in the Developer tool editor.

Creating an XMap

Create an XMap on the Data Transformation **Objects** view. When you create an XMap, you must have a schema that describes the input and the output hierarchal documents. You select the element in the schema that is the root element for the input hierarchy.

1. In the Data Processor transformation **Objects** view, click **New.**
2. Select XMap and click **Next.**
3. Enter a name for the XMap.
4. If the XMap component is the first component to process data in the transformation, enable **Set as startup component**.

   Click **Next.**
5. If you choose to create a schema reference, select **Add reference to a Schema Object**, and browse for the Schema object in the Model repository.

   To import a new Schema object, click **Create a new schema object**.
6. If you have a sample hierarchical file that you can use to test the XMap with, browse for and select the file from the file system.

   You can change the sample hierarchical file.
7. Choose the root for the input hierarchy.

   In the **Root Element Selection** dialog box, select the element in the schema that is the root element for the input hierarchal file. You can search for an element in the schema. You can use pattern searching. Enter `*<string>` to match any number of characters in the string. Enter `?<character>`to match a single character.
8. Click **Finish**.

   The Developer tool creates a view for each XMap that you create. Click the view to configure the mapping.

Create the Ports

Configure the input and output ports in the **Overview** view.

When you configure additional input or output ports in a Script, the Developer tool adds additional input ports and additional output ports to the transformation by default. You do not add input ports on the **Overview** view.

1. Define the input port datatype, port type, precision and scale.

2. Define the output port datatype, port type, precision, and scale.
3. If a Script has additional input ports, you can define the location of the example input file for the ports. Click the **Open** button in the **Input Location** field to browse for the file.

Deploy the Data Transformation Service to the Data Transformation Repository

To make a Data Transformation Service available to run with HParser commands, copy the service to the HParser nodes.

Export the Data Processor transformation as a Data Transformation service. After you export the Data Transformation service from the Developer tool, copy the service from the ServiceDB on the Data Transformation machine to the ServiceDB on the HParser command node. Then copy the service to the ServiceDB on all nodes of the Hadoop cluster.

Create a Directory for the Input and Output Files

Create a directory for the input and output files on the command node.

Create the `tmp` directory on the command node:

```
hadoop fs -mkdir /<directory name>
```

Copy the source files for the Data Transformation service to the directory.

```
hadoop fs -put /var/lib/hparser/ServicesDB/testCME/hl7-obs.txt /tmp/hl7-obs.txt
```

Run the MapReduce Job

The MapReduce job contains one or more tasks that assign input data to available instances of a Data Transformation service. Submit a job from the command line of the command node.

MapReduce Command Syntax

The hadoop command runs a MapReduce job.

Run a MapReduce job from the command line of the machine you configure as the command node.

The Hadoop command uses the following syntax :

```
hadoop jar [HParserJARFile]  -D<ParameterName>=<Value> ] [-conf ConfigurationFileName]
InputPath OutputPath [ ServiceName ]
```

Separate each value by a space.

The following table describes Hadoop options and arguments:

Option	Argument	Description
jar	<HParserJARFile>	Required. The name of the Data Transformation HParser JAR file. For example: dt-hadoop-0.1.1.jar
-D	<ParameterName>=<Value>	Optional. Input parameters for the Data Transformation service. -D must precede the parameter without a space character. The parameter has the format **name=value**. If the value contains any space, enclose the parameter in double quotes. Multiple input parameters can appear in any order. For example: `-Ddt.service.name=HIPAA_5010A Parser`
-conf	<ConfigurationFileName>	Optional. The path and file name of an HParser configuration file. You can define a configuration file if you want to override the default HParser configuration parameters.
"InputPath"	n/a	Required. The path and file name of the source files. The path must be enclosed in double quotes and can contain a regular expression in the format supported by Hadoop. For example: `"/user/mike/input/837P/Parser/Test*/input[1-5].txt"`
OutputPath	n/a	Required. Path of the output file. For example: `/user/mike/output/output.txt`
ServiceName	n/a	Name of the Data Transformation service. For example: `HIPAA_5010A_Parser` Required when you do not define the name of the service with the -D option.

Input Parameters (-D)

Use the -D option to define the `MapReduce` job parameters for the Data Transformation service.

Multiple instances of the -D option can appear in any order and must be separated by spaces. You can use the -D option with other parameters defined in the Hadoop specification. For more information, see the Hadoop documentation.
Note: The MapReduce command fails if you enter a space character between -D and the argument.

The following table describes the parameters you can use with the -D option:

Parameter Name	Description
dt.debug	Enable HParser debug logging to the Hadoop task log. Enter true or false. Default is false. The parameter datatype is Boolean.
dt.deploy.hdfs.home	HDFS directory name where HParser can find the license file and services when `dt.deploy.service` or `dt.deploy.license` is enabled. Default is `/user/hparser`. The parameter datatype is String.

Parameter Name	Description
dt. deploy.license	When you enable `dt.deploy.license`, HParser uses the Data Transformation license file from HDFS. Enter true or false. Default is false. The parameter datatype is Boolean.
dt.deploy.service	When you enable `dt.deploy.service`, HParser looks for the Data Transformation service in the following HDFS directory: `<dt.deploy.hdfs.home>/ServiceDB`. Enter true or false. Default is false. The parameter datatype is Boolean.
dt.service.name	The name of the Data Transformation service. For example: `-Ddt.service.name=HIPAA_5010A_Parser` Required when you do not define the name of the service as the last option for the hadooop command. The parameter datatype is String.

dt.service.parameter.<VariableName>	The value of a variable that you defined in the Data Transformation service. For example:
	`-Ddt.service.parameter.MyVariable1=50`
	`-Ddt.service.parameter.MyVariable2="Hello, World."`
	The parameter datatype must be of the same type as the variable that you defined in the Data Transformation service.

dt.output.hbase.colNames	List of column names to store in an HBase table. The list matches the order of the fields in the Data Transformation service output. For example: `provider_name:provider_address:patient_dob` The parameter datatype is String.
dt.output.hbase.columnFamilyName	HBase column family name. The parameter datatype is String.
dt.output.hbase.enabled	Determines whether you can store the Data Transformation service in an HBase table. Enter true or false. Default is false. The parameter datatype is Boolean.
dt.output.hbase.tableName	HBase table name. The parameter datatype is String.
dt.output.rootElement	When the Data Transformation service returns an XML document, you can enclose the output elements with a root element to create a valid XML document that can be read by an XML parser. You can configure the root element name. For example, if you configure `dt.output.rootElement=root`, Hparser returns <root> at the beginning of the XML file and it returns </root> at the end of the XML file. The parameter datatype is String.
dt.output.timestamp	Create the output directory with a timestamp in the directory name. The timestamp is in the following format: `yyyy.MM.dd.HH.mm.ss`. For example, the output directory might have the following path: `/tmp/out_2012.10.03.10.55.34`. Enter true or false. Default is false. The parameter datatype is Boolean.
dt.output.row.delimiter	The delimiter that Hadoop inserts between output segments received from the Data Transformation service. Hadoop writes the delimiter to the output file in UTF-8 encoding. For example: `-Ddt.output.row.delimiter=@@@` Default is newline (\n). The parameter datatype is String.

Parameter Name	Description
dt.output.drop.newlines	A Boolean value that determines whether the output of the Data Transformation service contains newline characters. Enter one of the following values: - true. Remove newline characters from the output of the Data Transformation service. - false. The output of the Data Transformation service is unchanged. For example: `-Ddt.output.drop.newlines=true` Default is true. The parameter datatype is Boolean.
dt.streamer.chunk.size	The size, in bytes, of the chunks passed to the Data Transformation streamer service. If the service is not a streamer, this parameter is ignored. For example: `-Ddt.streamer.chunk.size=100000` The parameter datatype is a positive integer.
mapreduce.inputformat.class	Defines the input format for the Data Transformation service. For input types, see "Define the Input Format" on page 23. The parameter datatype is String.
mapreduce.map.class	To run Data Transformation as a MapReduce mapper, set this property equal to com.informatica.b2b.dt.hadoop.mappers.DTMapperDefault. HParser runs Data Transformation as a record reader by default. The parameter datatype is String.

Define the Input Format

To define the input format for the Data Transformation service, use the `InputFormat` class and the class options with the `MapReduce` job parameter.

Use an `InputFormat` class option with the `-D` option in the `MapReduce` job. For more information, see the Hadoop documentation.

The following table describes the options that you can use with the `InputFormat` class:

Parameter Name	Description
DTInputFormat	Defines the input format as a small file.
DTSplitInputFormat	Defines the input format as text delimited by a special character. Data Transformation can process each record in the file, or batches of records. Use the `DTSplitInputFormat` class option with the `dt.input.delimiter` option to define the delimiter. Add a delimiter definition with the following syntax to the XML document: `dt.input.delimiter=<string>`
DTStreamInputFormat	Defines the input format as a complex file with a dynamic delimiter that requires that Data Transformation provide the start of the record. Data Transformation can use a service to return the offset to the next record. Use the `DTtreamInputFormat` option with the `dt.record.service.name` option to define the service that returns the next record offset.

Parameter Name	Description
HBaseInputFormat	Defines the input format as an HBase table. Data Transformation retrieves data from a specified HBase table.
RegexInputFormat	Defines the input format as file records that are read using a regular expression. Use the `RegexInputFormat` class option with the following options: - dt.record.regex: Defines the regular expression. The default value is **\n**. - dt.record.encdoing : Defines the encoding. The default value is **UTF-8**. - **dt.record.skipheader**: Defines whether to skip the file header. The default value is **false**. To skip the file header, set to **true**. A sample HParser call is as follows: `hadoop jar /home/hadoop/dt-hadoop-1.1-cdh4.jar com.informatica.b2b.dt.hadoop.DataTransformationJob -Dmapreduce.inputformat.class=com.informatica.b2b.dt.hadoop.io.RegexInputFormat -Ddt.record.regex=@ -Ddt.record.skipheader=true /testDir/input.txt /testDir/outputFolder dtService`
SequenceFileInputFormat	Defines the input format as a sequence file. Data Transformation processes each record in a sequence file.
TextInputFormat	Defines the input format as text delimited by the newline delimiter. Data Transformation can process each line in the file, or batches of lines.
XmlInputDelimiter	Defines the input format as XML that is delimited by an XML delimiter. Use the `xml.input.delimiter` tag to define record boundaries. For example, if you configure `xml.input.delimiter=dept` for the XML document "<org> <dept> <emp> <emp> … </dept> <dept> … </dept> … </org>" , each map task receives a `<dept>` element.

InputFormat Examples

To run HParser with sequence file input, use the following `InputFormat` class option syntax:

```
-Dmapreduce.inputformat.class=org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat
```

To run HParser with XML input, use the following `InputFormat` class option syntax:

```
-Dmapreduce.inputformat.class=com.informatica.b2b.dt.hadoop.io.XmlInputFormat
```

Mapper Options

To define how the HParser fetches hadoop records for the Data Transformation service, use a mapper option with the `MapReduce` job parameter. The mapper option defines how HParser filters and sorts hadoop records to obtain the input data for the Data Transformation service.

Use a mapper option with the `-D` option in the `MapReduce` job. For more information, see the Hadoop documentation.

The following table describes the options that you can use to define the HParser map operation:

Parameter Name	Description
DTMapper	Defines the map format as records or batches. Data Transformation processes each record or batch.
passthrough	The data is passed through to the output without any transformation.

Reducer Options

To define how the HParser recombines hadoop records that it has fetched as input for the Data Transformation service, use a reduce option with the `MapReduce` job parameter. The reduce option defines how HParser combines the sorted and filtered hadoop records to obtain the input data for the Data Transformation service.

Use the reduce option with the `-D` option in the `MapReduce` job. For more information, see the Hadoop documentation.

The following table describes the options that you can use to define the HParser reduce operation:

Parameter Name	Description
DTReduce	Defines the data as records or batches with the same key. Data Transformation processes records or batches with the same key. When you use the `DTReduce` option with the `InputFormat` class and the `HBase` class option, the output is written to HBase.
passthrough	The data is passed through to the output without any transformation.

Output Options

You can configure the output format for the Data Transformation service with direct settings or with the settings for the mapper and reducer.

The following table describes the options that you can use to define the output:

Output Format	Description
AVRO	To define the output format as an AVRO file, use the AVRO output option in the `MapReduce` job with the following syntax: `avro.apache.org.AvroOutputFormat`
File	By default, the HParser reducer option `DTReduce` produces a single file as output.
HBase	To obtain HBase output, use the `InputFormat` class with the `HBase` class option, and the `DTReduce` reducer option in the `MapReduce` job.
Multiple Files	For each mapper or reducer transformation output, designate a target file for each record.

Troubleshooting

Error messages appear in the console where you run Hadoop.

Messages also appear in the following logs:

- The Data Transformation event log for each node in the cluster
- The Hadoop job log

- The Hadoop task log
- The Hadoop HDFS log

If a problem occurs during task processing, HParser writes error messages to the Data Transformation event log and MapReduce writes error messages to the Hadoop task log. At the end of the job, MapReduce writes a success or failure message to the Hadoop job log.

You can also view additional messages when you set the option `dt.debug=true` in the Hadoop command.

HParser Error Messages

The following list describes the HParser error messages:

The '<JobName>' job has failed. Check the log files to determine the cause.

Less than the required percentage of tasks in the job have succeeded. Some tasks might have completed.

The JRE was unable to process <encodingName> encoding. Check the log files to determine the cause.

The JRE was unable to process UTF-8 encoding.

The host name of a node cannot be identified. Reason: <message>.

The host name of a node cannot be identified. Java returns a diagnostic message.

Warning: "<size>" is not a valid streamer chunk size. Data Transformation is trying to run the task with the default size (<DefaultSize>). Set `dt.streamer.chunk.size` to a positive integer.

dt.streamer.chunk.size was not a positive integer. Data Transformation used a default value instead.

Error reading the input. Reason: <message>.

Verify that the HDFS file system is running correctly, and then run the job again. Check the Hadoop HDFS log and task log.

The <fileName> file is empty.

Check the input file or remove it from the input directory.

A task has failed. Check the log files and refer to the following information:

File: <FilePath>

Host: <HostName>

Problem: <FailedTaskOperationDescription>

Corrective action: <CorrectiveAction>

Failure report from Data Transformation:

<DataTransformationFailureReport>

Task Operation Messages

The following list describes messages for failed task operations:

The input data caused the Data Transformation streamer to fail.

Check the Data Transformation event log and the Data Transformation service.

The streamer failed while returning output to Hadoop.

Check the Data Transformation event log and the Data Transformation service.

The Data Transformation streamer returned an error message.

Check the Data Transformation event log.

Note: For more information about the Hadoop logs, see the Hadoop documentation. For more information about Data Transformation event logs, see the *Data Transformation Administrator Guide*.

CHAPTER 4

Example

This chapter includes the following topics:

- Example Overview, 28
- Step 1. Copy the Data Transformation Service to the Data Transformation Repository, 28
- Step 2. Create a Directory for the Input and Output Files, 29
- Step 3. Run the MapReduce Job, 29

Example Overview

For this example, you can use a Data Transformation service that installs with HParser. The service is a Parser transformation called `testCME`. You can use the testCME service to test the HParser setup. The service runs on an Apache Hadoop 0.21 cluster.

The `testCME` parser processes an HL7 message having the following structure:

```
MSH|^~\&|LAB||CDB||||ORU^R01|K172|P
PID|||PATID1234^5%M11||Jones^William||19610613|M
OBR||||80004^Electrolytes
OBX|1|ST|84295^Na||150|mmol/l|136-148|Above high normal|||Final results
OBX|2|ST|84132^K+||4.5|mmol/l|3.5-5|Normal|||Final results
OBX|3|ST|82435^Cl||102|mmol/l|94-105|Normal|||Final results
OBX|4|ST|82374^CO2||27|mmol/l|24-31|Normal|||Final results
```

The output of the parser is a message saying `Test Succeeded`.

Step 1. Copy the Data Transformation Service to the Data Transformation Repository

Copy the testCME service to the Data Transformation repository. The repository is a folder called `ServiceDB`.

▶ Copy the `testCME` folder from the **setupTests/testCME** directory to the **ServiceDB** directory.

cp -R /usr/lib/hparser/setupTests/testCME /var/lib/hparser/ServicesDB/testCME

Step 2. Create a Directory for the Input and Output Files

Create a directory for the input and output files on the command node.

1. Create the `tmp` directory on the command node:

   ```
   hadoop fs -mkdir /tmp
   ```

2. Copy the sample input file from the Data Transformation repository to the `tmp` directory. The sample input file is `tmp/hl7-obs.txt`.

   ```
   hadoop fs -put /var/lib/hparser/ServicesDB/testCME/hl7-obs.txt /tmp/hl7-obs.txt
   ```

Step 3. Run the MapReduce Job

Run the MapReduce job from the command line of the command node.

1. To submit the MapReduce job, enter the following text on the command line:

   ```
   hadoop jar hparser-1.1-hadoop-0.2.1 -conf hparser_conf.xml /tmp/hl7-obs.txt /tmp/out
   testCME
   ```

   Note: The JAR file must be hparser-1.1-hadoop-0.2.1 to run on Apache Hadoop 0.21. The configuration file is optional.

2. View the `out.xml` file in the `/tmp` directory:

   ```
   <Result> Test Succeeded </Result>
   ```

INDEX

C

command node
  configuring HParser 14

D

-D command option
  MapReduce options 21
Data Processor transformation
  creating 17
Data Transformation Service
  deploying 20
delimiter between output records
  MapReduce job 21
deploy
  Data Transformation Service 20

H

Hadoop command
  syntax and options 20
HParser
  configuring command node 14
  directory permissions 13
  example 28
  input and outpt directories 20
  installing 13
  running on Hadoop overview 16
  troubleshooting 25
  uninstalling 15

I

input formats
  MapReduce job 23
InputFormat class option
  MapReduce options 23

M

Map
  MapReduce options 25
Mapper
  MapReduce options 24
MapReduce job
  -D option 21
  command syntax 20
  InputFormat class 23
  Mapoptions 25
  Mapper options 24
  Output class 25
  running 20

O

output
  MapReduce job 24, 25
output formats
  MapReduce job 25
Output option
  MapReduce options 25

S

sample source file
  defining in Data Processor transformation 18
Script
  creating in Data Processor transformation 18

T

troubleshooting
  HParser 25

X

XMap object
  creating in Data Processor transformation 19