Information Builders

iWay

iWay Big Data Integrator User's Guide

Version 1.5.2

August 21, 2018

Active Technologies, EDA, EDA/SQL, FIDEL, FOCUS, Information Builders, the Information Builders logo, iWay, iWay Software, Parlay, PC/FOCUS, RStat, Table Talk, Web390, WebFOCUS, WebFOCUS Active Technologies, and WebFOCUS Magnify are registered trademarks, and DataMigrator and Hyperstage are trademarks of Information Builders, Inc.

Adobe, the Adobe logo, Acrobat, Adobe Reader, Flash, Adobe Flash Builder, Flex, and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2017, by Information Builders, Inc. and iWay Software. All rights reserved. Patent Pending. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Contents

Preface 9
Documentation Conventions 10
Related Publications 11
Customer Support 11
Help Us to Serve You Better 12
User Feedback 14
Information Builders Consulting and Training 14
1. Introducing iWay Big Data Integrator 15
iWay Big Data Integrator (iBDI) Overview 15
Platform Terminology. 15
iWay Big Data Integrator Supported Vendors 15
iWay Big Data Integrator Key Features 16
2. iWay Big Data Integrator Quick Start Guide 17
iWay Big Data Integrator Quick Start Overview 17
iBDI User Types 17
Big Data Integrator Client Installation Considerations 19
Big Data Integrator Server Deployment Considerations 19
Big Data Integrator Use Case Scenarios 19
3. Installing iWay Big Data Integrator 21
Big Data Integrator Prerequisites 21
JDBC Drivers. 21
Server Deployment. 21
Client Workbench. 22
Installing iWay Big Data Integrator 22
Allocating Java Memory to iWay Big Data Integrator 27
Configuring Secure Connections With Kerberos 28
4. Getting Started With iWay Big Data Integrator 29
Understanding the Layout and User Interface 29
Understanding the Components of the Database Connections Facility 31
Connecting to a Data Source 37
Creating a New Project in iBDI 62

Project Components. 67

5. Defining Data in the Data Source Explorer 73

Using the Data Source Explorer for Extract, Transform, and Load Operations 73

Understanding the Data Submenu Options. 73

Edit. 74

Sample Contents. 74

Extract. 74

Load. 76

Replace. 77

Using the Generate DDL Wizard 77

6. Understanding Pipelines 85

Understanding Apache Spark 85

Understanding a Pipeline 85

Source Stages. 86

Compute Stages. 86

Target Stages. 88

Using the Pipeline Builder 88

Opening the Pipeline Builder. 89

Adding, Deleting, and Modifying a Stage. 90

Adding, Deleting, and Modifying a Source. 90

Using the Problems Pane. 91

Chaining Pipelines. 92

Troubleshooting Pipelines. 92

Building Your First Pipeline 93

Configuring Apache NiFi 94

7. Understanding the Transformer 99

iBDI Transformer Overview 99

Getting Started With the iBDI Transformer 99

Mapping Fields to the Output Schema. 105

Complex Mappings. 106

Joining Tables. 106

General Expression Builder. 106

Where Clause Expression Builder. 107
HQL. 107
Saving Your Work. 107

8. Creating a Sqoop Configuration 109

Creating and Executing a Sqoop Configuration 109
Exporting Data Using Sqoop 113

9. Creating a Flume Configuration 115

Understanding Flume 115
Creating and Executing a Flume Configuration 115
Flume Parameters Reference 119
Common Parameters. 119
Source Parameters. 120
Avro. 120
HTTP. 120
Exec. 120
SpoolDirSource. 120
ExecSource. 121
JMSSource. 121
Channel Parameters. 122
Memory Channel. 122
File Channel. 122
Sink Parameters. 122
Avro Sink. 123
Kafka Sink. 123
HDFS Sink. 123
Flume File Name Escape Sequences. 125

10. Creating a Mapping Configuration 127

Big Data Mapper Overview 127
Creating and Configuring a New Mapping 128

11. Creating a Wrangler Configuration 139

Configuring a Connection to a Hadoop Distributed File System Server 139
Creating a Wrangler Configuration 143

12. Defining Run Configurations 147
iBDI Runtime Configuration Considerations 147
Defining Run Configurations for a Pipeline 148
Defining Run Configurations for a Sqoop or a Flume 152
Defining Run Configurations for a Spark Deployment 156

13. Pipeline Parameters Reference 159
Pipeline Source Type Parameters 159
HDFS. 159
Hive. 160
HQL. 160
Stream. 160
Kafka. 161
Kafka Direct. 162
Nifi. 162
RDBMS. 163
Pipeline Compute Type Parameters 164
Cleanse. 164
Coalesce. 164
Distinct. 164
Drop. 164
Filter. 165
Hive. 165
HQL. 165
RStat Model. 165
Order By. 165
Sample. 165
Save. 166
Select. 167
Show. 167
Pipeline Target Types 167
Console. 167
HDFS. 167

HDFS CSV. 168
Kafka. 168
RDBMS. 168
Data Format File Types 169
Streaming Considerations 170
Streaming Data Structure. 170
Defining a Structure Using Avro Schema. 170
Data Types. 170
XML and JSON Data Streams. 170
Nifi Streaming. 171

14. Configuring Kerberos Authentication 173

Creating the Java Authentication and Authorization Service Login Configuration File 173
Configuring Kerberos Authentication on Windows 174
Download and Install MIT Kerberos for Windows. 174
Using MIT Kerberos Ticket Manager to Get Tickets. 174
Using the Driver to Get Tickets. 175
Configuring the Kerberos Configuration File. 176
Configuring Kerberos Authentication on Linux 176
Configuring Kerberos Authentication on Mac OS X 177

A. Glossary 183

Preface

iWay Big Data Integrator (iBDI) is a solution that provides a distributed design and execution framework, which enables the power of Hadoop and Spark as a data integration and analysis platform. Using iBDI, you can import and transform data *natively* in Hadoop and Spark. In the Integration Edition, Spark pipelines enable powerful combinations of sources and compute functions to import and transform data. iBDI also provides relational data replication and Change Data Capture (CDC) using Apache Sqoop along with data de-duplication. iBDI also provides streaming and unstructured data capture using Apache Flume. In addition, iBDI also provides *native* data transformation capabilities with a rich array of out of the box functions. This documentation describes how to install, configure, and use iBDI.

How This Manual Is Organized

This manual includes the following chapters:

	Chapter/Appendix	Contents
1	Introducing iWay Big Data Integrator	Provides an introduction to iWay Big Data Integrator (iBDI).
2	iWay Big Data Integrator Quick Start Guide	Provides a quick start guide to iWay Big Data Integrator (iBDI).
3	Installing iWay Big Data Integrator	Provides prerequisites and describes how to install iWay Big Data Integrator (iBDI).
4	Getting Started With iWay Big Data Integrator	Provides an overview of the iWay Big Data Integrator (iBDI) design-time environment, describes key facilities, and tasks.
5	Defining Data in the Data Source Explorer	Describes how to define data in the Data Source Explorer and use the Generate DDL wizard to export a model from any RDBMS for iWay Big Data Integrator (iBDI) to translate to HQL and create the tables in Hive.
6	Understanding Pipelines	Introduces pipelines and describes how to configure pipelines using iWay Big Data Integrator (iBDI).
7	Understanding the Transformer	Introduces the Transformer and describes how to configure transforms using iWay Big Data Integrator (iBDI).

	Chapter/Appendix	Contents
8	Creating a Sqoop Configuration	Introduces Sqoops and describes how to create a Sqoop configuration using iWay Big Data Integrator (iBDI).
9	Creating a Flume Configuration	Introduces Flumes and describes how to create a Flume configuration using iWay Big Data Integrator (iBDI).
10	Creating a Mapping Configuration	Introduces mappings and describes how to create a mapping configuration using the Big Data Mapper in iWay Big Data Integrator (iBDI).
11	Creating a Wrangler Configuration	Introduces wranglers and describes how to create a wrangler configuration in iWay Big Data Integrator (iBDI).
12	Defining Run Configurations	Describes how to define run configurations using iWay Big Data Integrator (iBDI).
13	Pipeline Parameters Reference	Provides a reference for the available pipeline parameters.
14	Configuring Kerberos Authentication	Describes how to configure Kerberos authentication.
A	Glossary	Provides a reference for common terms that are used in Big Data discussions and iWay Big Data Integrator (iBDI).

Documentation Conventions

The following table describes the documentation conventions that are used in this manual.

Convention	Description
THIS TYPEFACE or this typeface	Denotes syntax that you must enter exactly as shown.
this typeface	Represents a placeholder (or variable), a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option that you can click or select.

Convention	Description
underscore	Indicates a default setting.
Key + Key	Indicates keys that you must press simultaneously.
{ }	Indicates two or three choices. Type one of them, not the braces.
|	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis (...).
. . .	Indicates that there are (or could be) intervening or additional commands.

Related Publications

Visit our Technical Documentation Library at *http://documentation.informationbuilders.com*. You can also contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have any questions about this product?

Join the Focal Point community. Focal Point is our online developer center and more than a message board. It is an interactive network of more than 3,000 developers from almost every profession and industry, collaborating on solutions and sharing tips and techniques. Access Focal Point at *http://forums.informationbuilders.com/eve/forums*.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our website, *http://www.informationbuilders.com*. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of *http://www.informationbuilders.com* also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

Call Information Builders Customer Support Services (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code number (*xxxx.xx*) when you call.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Help Us to Serve You Better

To help our consultants answer your questions effectively, be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following tables list the environment information our consultants require for the Client platform.

Client Platform	
Operating System	
OS Version	
JVM Vendor	
JVM Version	

The following tables list the environment information our consultants require for the Server platform

Server Platform	
Operating System	
OS Version	
JVM Vendor	
JVM Version	

The following table lists iBDI-related information required by our consultants.

Component	Client or Server
Run time object	Flume, Sqoop, Pipeline, etc.
Pipeline stages	❑ Source(s) ❑ Compute(s) ❑ Target(s)
System logs or iBDI console logs	

The following table lists additional questions to help us serve you better.

Request/Question	Error/Problem Details or Information
Provide usage scenarios or summarize the application that produces the problem.	
When did the problem start?	
Can you reproduce this problem consistently?	
Describe the problem.	
Describe the steps to reproduce the problem.	
Specify the error message(s).	
Any change in the environment: software configuration, database configuration, application, and so forth?	
Under what circumstance does the problem *not* occur?	

User Feedback

In an effort to produce effective documentation, the Technical Content Management staff welcomes your opinions regarding this document. Please use the Reader Comments form at the end of this document to communicate your feedback to us or to suggest changes that will support improvements to our documentation. You can also contact us through our website, *http://documentation.informationbuilders.com/connections.asp*.

Thank you, in advance, for your comments.

Information Builders Consulting and Training

Interested in training? Information Builders Education Department offers a wide variety of training courses for this and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our website (*http://education.informationbuilders.com*) or call (800) 969-INFO to speak to an Education Representative.

Chapter 1

Introducing iWay Big Data Integrator

This chapter provides an introduction to iWay Big Data Integrator (iBDI) and describes key features and facilities.

In this chapter:

- iWay Big Data Integrator (iBDI) Overview
- iWay Big Data Integrator Supported Vendors
- iWay Big Data Integrator Key Features

iWay Big Data Integrator (iBDI) Overview

iWay Big Data Integrator (iBDI) is an a integrated design and execution framework, enabling the power of Spark and Hadoop as a data integration, transformation and storage platform. iBDI can ingest data from traditional data stores, streaming sources, and other big data platforms. iBDI can then match, merge, or transform this data for storage, in memory computing, or streaming output. iBDI unlocks the potential of a big data platform without writing custom code or requiring a complex configuration.

Platform Terminology

This documentation provides a glossary, which defines common terms for the Hadoop platform and the Spark cluster computing platform. If these terms are unfamiliar to you while you read through this documentation, see *Glossary* on page 183.

iWay Big Data Integrator Supported Vendors

This section provides a summary of vendors that are supported by iWay Big Data Integrator (iBDI).

- iBDI is supported with Cloudera.
- iBDI works with Hortonworks and open source.

  Note: MapR is currently under research and is not fully warranted to function with iBDI version 1.5.2.

- iBDI is supported with the following vendor databases:
  - Postgres

- MySQL
- Microsoft SQL Server

iWay Big Data Integrator Key Features

iWay Big Data Integrator (iBDI) enables the following natively in a Hadoop or Spark ecosystem:

- Spark pipelines enable sequences of processes for obtaining, transforming, and persisting data in multiple combinations. Pipeline chains allow for complex processing without the need to write custom code.
- Change Data Capture (CDC) is unique to iBDI. iBDI provides extended Hadoop native tools to support CDC for limit file transfer sizes when possible.
- iBDI provides two modes of deployment:
  - *Deployment* for interactive runtime sessions.
  - *Publish* for batch and streaming sessions.
- iBDI provides the ability to transform data (Spark pipelines) through the Transformer tool or in native HDFS using the Mapper tool.
- Metadata can be extracted from iBDI-provisioned deployments in JSON format that can be joined with one another to describe:
  - Where the data originated.
  - Where the data was sent.
  - How the data was processed.
- iBDI allows you to:
  - Import data into Spark or Hadoop from wherever it exists (database, stream, Kafka, file, or other sources).
  - Provide metadata so that the data can be consumed by Spark or native Hadoop modules.
  - Transform data so that it can be consumed by other enterprise applications.

Chapter 2

iWay Big Data Integrator Quick Start Guide

This chapter provides a quick start guide to iWay Big Data Integrator (iBDI).

In this chapter:

- iWay Big Data Integrator Quick Start Overview
- iBDI User Types
- Big Data Integrator Client Installation Considerations
- Big Data Integrator Server Deployment Considerations
- Big Data Integrator Use Case Scenarios

iWay Big Data Integrator Quick Start Overview

This quick start guide summarizes the key high-level steps that are required to install, configure, and use iWay Big Data Integrator (iBDI) version 1.5.2. This quick start guide does not elaborate on any of the steps in detail. Instead, cross-references are provided for the corresponding sections within the documentation.

Users of iBDI are encouraged to follow the sequence of steps in this guide to quickly connect to Spark or Hadoop and begin using the components to connect to data sources. To gain a complete understanding about iBDI, it is recommended for users to review the entire *iWay Big Data Integrator User's Guide*, since this quick start guide is not a replacement for that level of detail.

iBDI User Types

This section provides a summary of the various user groups that iBDI can support, including some of their unique characteristics

- **Data Warehouse User**
  - Traditional and new sources to Hadoop and other storage systems.
  - Interested in data storage, organization, and retrieval.

- **Data Engineer**
  - Pattern analysis.
  - Data insight.
  - RStat models.
  - What-if scenarios.
- **High Speed Analytics User**
  - In memory, column-oriented computing.
  - New forms of storage.
  - Quickly retrieve business insights from data.
  - Identify trends and purchasing analysis.
- **Data Creators**
  - Require tools to merge, join, and manipulate tables and data.
  - Summarize existing data into new categories with additional data.
  - Expand existing data in combination with new data for expansive analysis.
- **Event Streamers**
  - Clickstream data and analytics.
  - Continuous flow of data.
  - Responsible for the following types of event handling:
    - **Detection.** Identify activity and events as they occur.
    - **Partitioning.** Split events into multiple parts.
    - **Enrichment.** Add information to event data.
    - **Aggregation.** Combine data from multiple events.
    - **Correlation.** Correlate events with other data.
    - **Storage.** Store event messages.
    - **Channel change.** Retrieve events in one format type and publish in another format.

Big Data Integrator Client Installation Considerations

- ❑ Ensure that you are using a supported environment, as described in *Client Workbench* on page 22.
- ❑ Ensure that sufficient memory is allocated for the Eclipse workbench, as described in *Allocating Java Memory to iWay Big Data Integrator* on page 27.
- ❑ Ensure that the correct JDBC drivers or other third-party drivers are available on your system. For more information on creating connections to data sources, see *Understanding the Components of the Database Connections Facility* on page 31 and *Connecting to a Data Source* on page 37.

Big Data Integrator Server Deployment Considerations

- ❑ Ensure that you are using a supported environment, as described in *Big Data Integrator Prerequisites* on page 21 (Hadoop Ecosystem Tools).
- ❑ Ensure that you have a valid deployment script and resource archive.
- ❑ Ensure that you have obtained a valid license for your deployment before installing iBDI.

Big Data Integrator Use Case Scenarios

- ❑ To configure secure connections between the client and the server, see *Configuring Kerberos Authentication* on page 173.
- ❑ For JDBC-based data, ensure that the proper drivers for the target data source are available in the client workspace, as described in *Understanding the Components of the Database Connections Facility* on page 31 and *Connecting to a Data Source* on page 37
- ❑ Performing a simple import and export:
  1. Create a new project, as described in *Creating a New Project in iBDI* on page 62.
  2. Create a Sqoop, as described in *Creating and Executing a Sqoop Configuration* on page 109 and *Exporting Data Using Sqoop* on page 113.
  3. Use a Flume for data in motion, as described in *Creating a Flume Configuration* on page 115.
  4. Run the configuration, as described in *Defining Run Configurations for a Sqoop or a Flume* on page 152.
- ❑ Configuring pipelines for Spark compound, multiple, or complex operations:
  1. Review *Understanding Apache Spark* on page 85 and *Understanding a Pipeline* on page 85.

2. Understand pipeline stages, as described in *Source Stages* on page 86, *Compute Stages* on page 86, and *Target Stages* on page 88.
3. Perform Hive transformations in pipelines, as described in *Understanding the Transformer* on page 99.
4. Review the available pipeline parameters and properties, as described in *Pipeline Parameters Reference* on page 159.
5. Use the Pipeline Builder to design and configure your pipelines, as described in *Using the Pipeline Builder* on page 88.
6. Run the configuration, as described in *Defining Run Configurations for a Pipeline* on page 148.

- Data Source Explorer for simple ETL and DDL. For more information, see *Defining Data in the Data Source Explorer* on page 73.
- Big table mappings for Hive table manipulations. For more information, see *Creating a Mapping Configuration* on page 127.
- Wrangler and HDFS-bridge unstructured to structured data. For more information, see *Creating a Wrangler Configuration* on page 139.

Chapter 3

Installing iWay Big Data Integrator

This chapter provides prerequisites and describes how to install iWay Big Data Integrator (iBDI).

In this chapter:

- Big Data Integrator Prerequisites
- Installing iWay Big Data Integrator
- Allocating Java Memory to iWay Big Data Integrator
- Configuring Secure Connections With Kerberos

Big Data Integrator Prerequisites

Review the prerequisites in this section before installing and using iWay Big Data Integrator (iBDI).

JDBC Drivers

JDBC drivers are not shipped with iBDI and must be obtained from a relevant JDBC vendor (for example, Cloudera and Simba). Depending on your usage requirements, there are several open source Hive JDBC drivers available online. For additional security and availability options, commercial drivers are also available. Each driver includes a list of dependencies for the specific driver. Consult the driver provider for the list of .jar files that must be used with the specific driver version.

Server Deployment

A Hadoop Edge node (also referred to as a gateway node) is the interface between a Hadoop cluster and an external network. An Edge node is used to run client applications and administration tools for clusters. An Edge node is often used as a staging environment for data that is being transferred into a Hadoop cluster.

The iBDI Hadoop or Spark environment must be *provisioned* for the product to function. This requires installing the iBDI runtime Java .jar files into a folder on the Edge node.

For more information, see the *iWay Big Data Integrator Provisioning Guide*.

Client Workbench

This section provides prerequisites for the iBDI client workbench.

- 8GB RAM on either Windows (Version 7, 8, or 10), Mac (OS X El Capitan), or Linux (Ubuntu or Centos).
- Java Version 8 (1.8_31 and higher).
- Download one of the following packaged archive files based on your operating system:
  - Linux: *bdi-1.5.2.I201702211956-linux.gtk.x86_64.tar.gz*
  - Mac OS X: *bdi-1.5.2.I201702211956-macosx.cocoa.x86_64.tar.gz*
  - Windows: *bdi-1.5.2.I201702211956-win32.win32.x86_64.zip*

  Note: This documentation references the Windows version of iBDI.
- Depending on the database, download the required JDBC driver and save the driver to your local system:
  - Postgres: *postgresql-9.4-1201.jdbc4.jar*
  - MySQL: *mysql-connector-java-5.1.28-bin.jar*
  - SQL Server: *sqljdbc4.jar*
  - Appropriate driver for Hadoop distribution.

Installing iWay Big Data Integrator

This section describes how to install iWay Big Data Integrator (iBDI) on Windows and Linux platforms.

Procedure: How to Install iWay Big Data Integrator on Windows

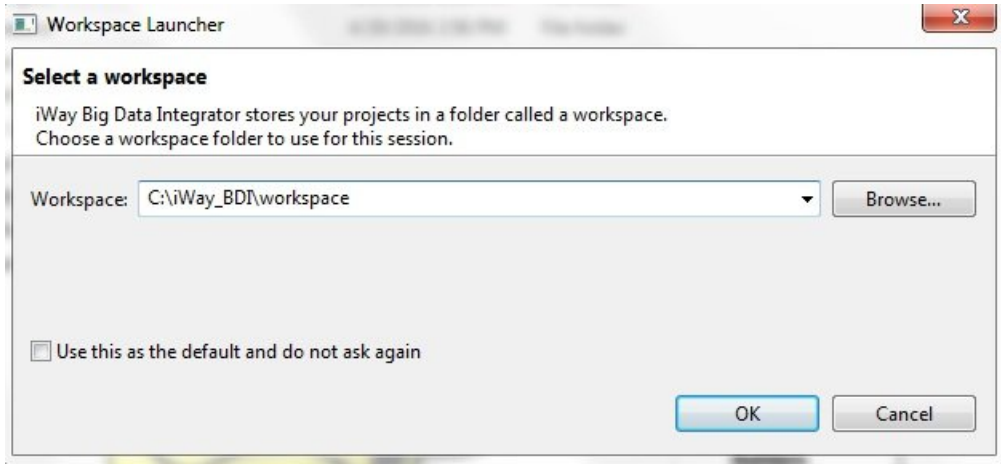
1. Extract the *bdi-1.5.2.I201702211956-win32.win32.x86_64.zip* file to the following folder:

   ```
   C:\iWay_BDI
   ```

2. After the .zip file is extracted, double-click the *iWayBigDataIntegrator.exe* file in the installation folder. For example:

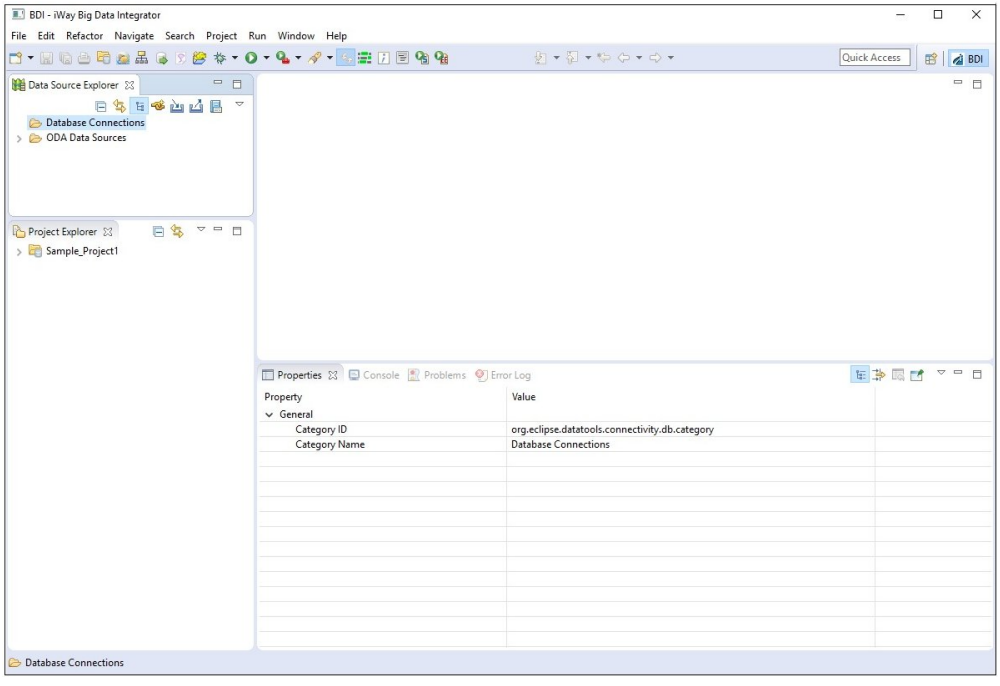
   ```
   C:\iWay_BDI\iWayBigDataIntegrator.exe
   ```

The Workspace Launcher dialog opens, as shown in the following image.



3. Specify a custom workspace according to your requirements or accept the default value.
4. Click *OK*.

iWay Big Data Integrator opens as a perspective in the Eclipse framework, as shown in the following image.



Procedure: How to Install iWay Big Data Integrator on Linux

Before continuing, ensure that an X Window System is available, which is a prerequisite.

1. Move the *bdi-1.5.2.I201702211956-linux.gtk.x86_64.tar.gz* file to a folder in the selected user to run iBDI using FTP, SFTP, or another copy method. For example:

   ```
   /home/bdiuser/install
   ```

2. Right-click the *bdi-1.5.2.I201702211956-linux.gtk.x86_64.tar.gz* file and select *Open with File Roller* or use another file extractor utility.

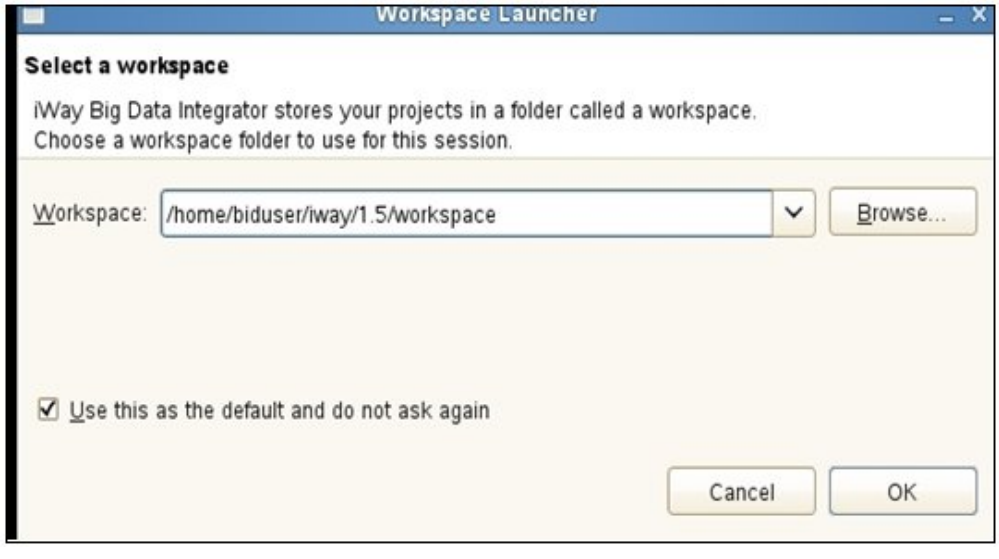
   Select an installation folder, for example:

   ```
   /home/bdiuser/bdi15
   ```

The iBDI extract process creates a folder structure, as shown in the following image.

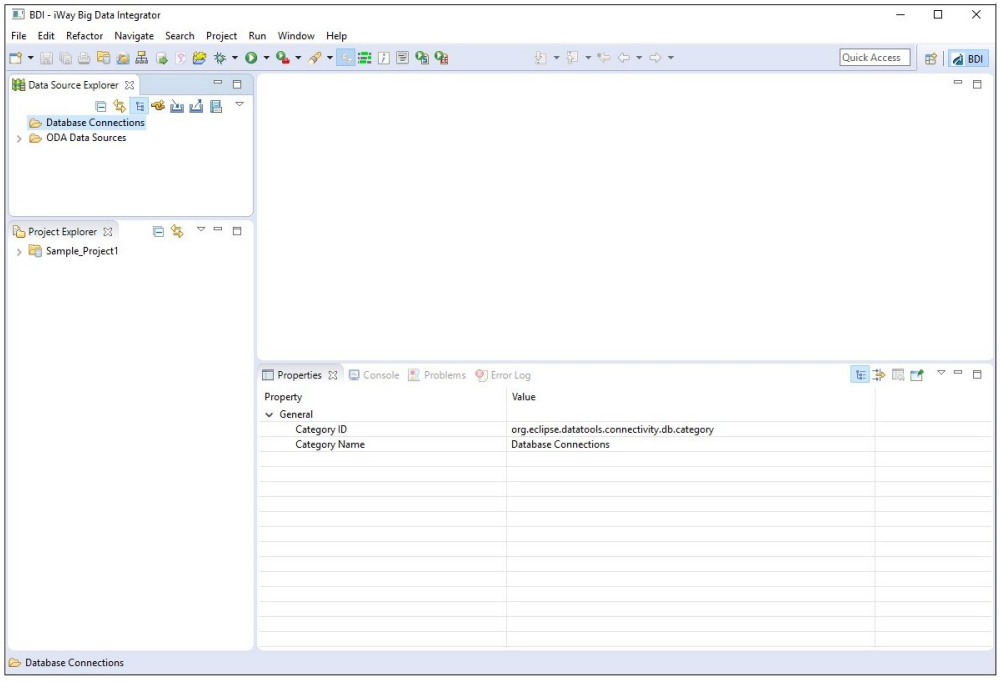
3. Double-click the *iWayBigDataIntegrator* icon.

   The Workspace Launcher dialog opens, as shown in the following image.



4. Specify a custom workspace according to your requirements or accept the default value.
5. Click *OK*.

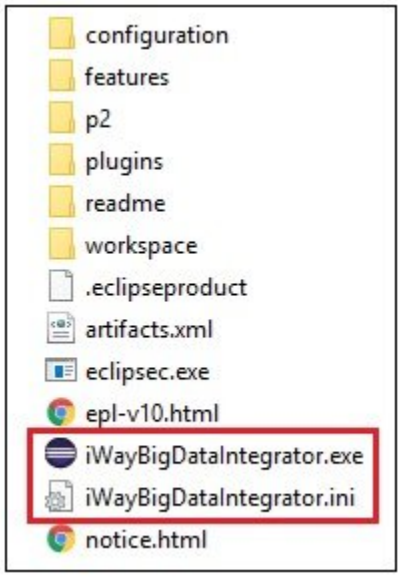
iWay Big Data Integrator opens as a perspective in the Eclipse framework, as shown in the following image.



For more information on each of the main areas available in this perspective, see *Understanding the Layout and User Interface* on page 29.

Allocating Java Memory to iWay Big Data Integrator

The *iWayBigDataIntegrator.ini* file resides in the same folder as the *iWayBigDataIntegrator.exe* file, which is the root folder where iWay Big Data Integrator (iBDI) is installed.



Only the properties that are specified in this section can be modified. The properties are specified according to the Eclipse properties format. For more information, see the following website:

https://wiki.eclipse.org/Eclipse.ini

The iBDI national language settings are:

```
-nl
en_us
```

Only use these settings if iBDI projects are very large and memory or performance issues have occurred. Ensure that the underlying operating system has the amount of free memory available as specified in the following parameters. The iBDI memory settings are:

```
-XX Garbage collector heap size

-XX:MaxPermSize=256M

-Xms       set initial Java heap size

-Xms512M
```

```
-Xmx       set maximum Java heap size

-Xmx2048M
```

For more information on the memory parameter definitions and memory tuning, see the Oracle Java Virtual Machine (JVM) documentation.

Configuring Secure Connections With Kerberos

You can configure your Kerberos setup so that you can use the MIT Kerberos Ticket Manager to get the Ticket Granting Ticket (TGT), or configure the setup so that you can use the driver to get the ticket directly from the Key Distribution Center (KDC). Also, if a client application obtains a Subject with a TGT, you can use that Subject to authenticate the connection.

For more information, see *Configuring Kerberos Authentication* on page 173.

Chapter 4

Getting Started With iWay Big Data Integrator

This chapter provides an overview of the iWay Big Data Integrator (iBDI) design-time environment, describes key facilities, and tasks.

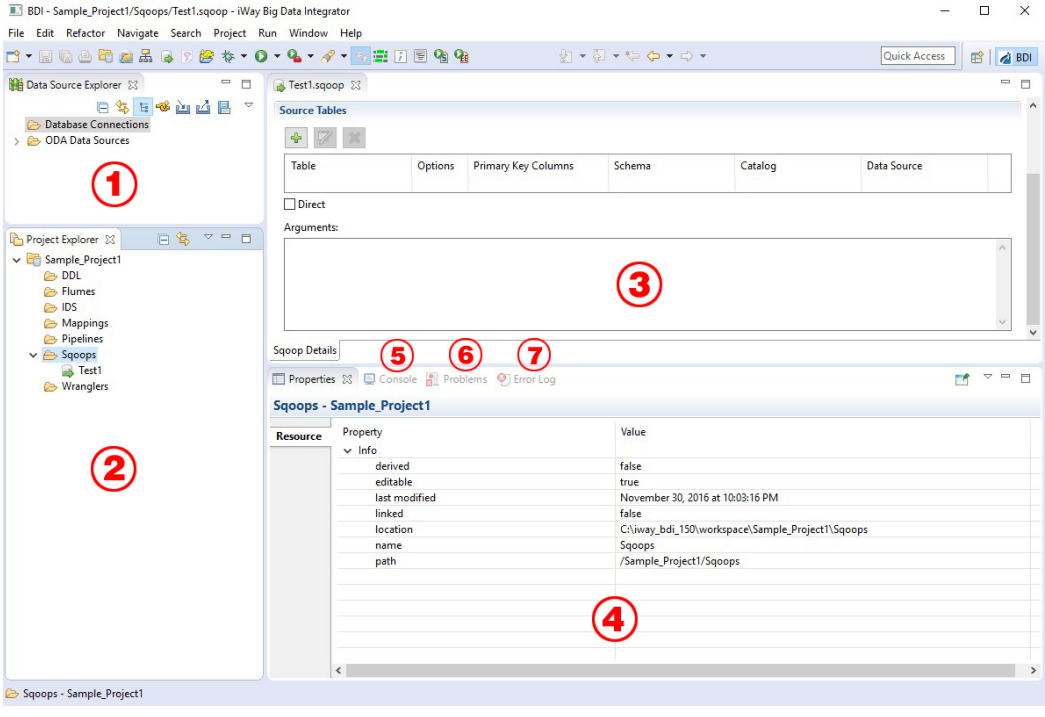
In this chapter:

- Understanding the Layout and User Interface
- Understanding the Components of the Database Connections Facility
- Connecting to a Data Source
- Creating a New Project in iBDI

Understanding the Layout and User Interface

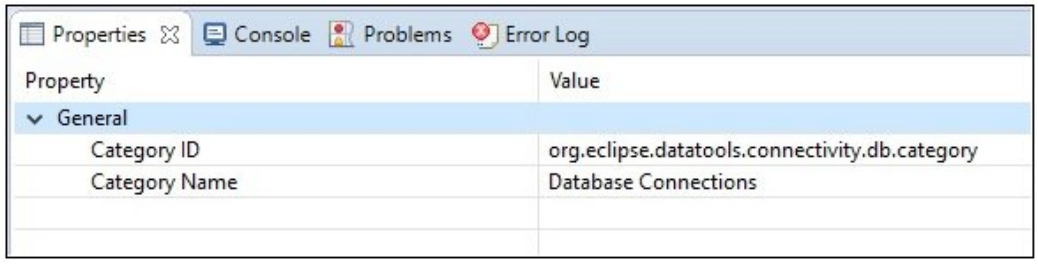
This section provides an overview of the iWay Big Data Integrator (iBDI) perspective.

The iBDI perspective is divided into four main areas, as shown in the following image.



1. **Data Source Explorer**, allows you to connect to a data source in iBDI (for example, Hive and PostgreSQL). You can also manage all of your data connections from this pane.
2. **Project Explorer**, allows you to create, manage, and deploy your projects in iBDI.
3. **Workspace**, represents the main working area in iBDI where you can build pipelines and view selected project components in more detail.
4. **Properties tab**, allows you to configure and view the properties (parameters) of any selected project asset or component.
5. **Console tab**, allows you to view the output of a selected job.
6. **Problems tab**, allows you to view issues and selected resolutions.

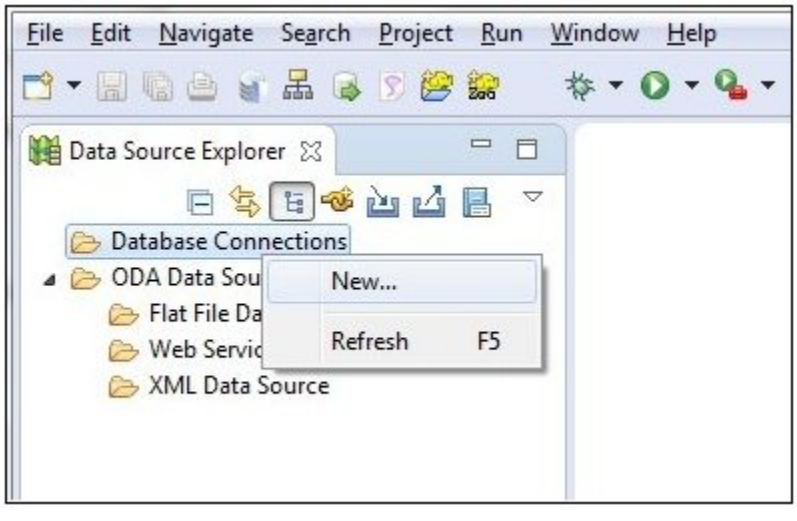
7. **Error Log tab**, allows you to view any job errors or build problems that may occur.



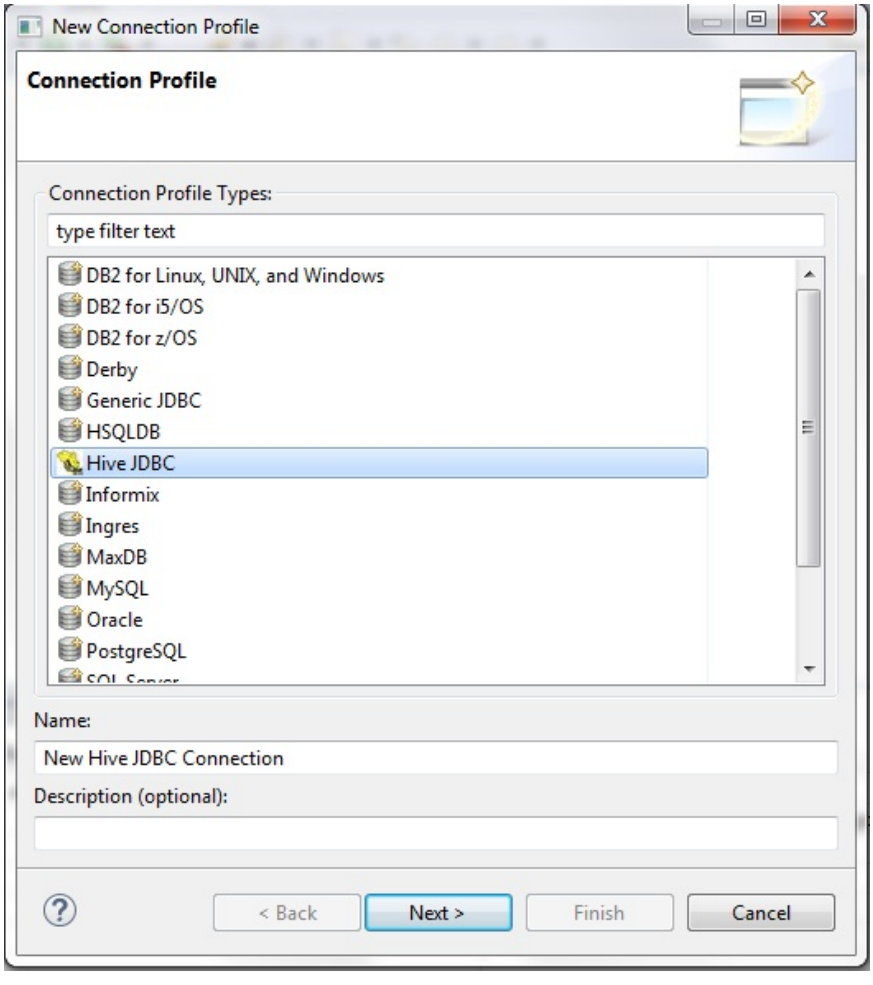
Understanding the Components of the Database Connections Facility

The Database Connections facility is composed of several dialogs that enable you to configure a connection to a data source (for example, Hive).

In the Data Source Explorer panel, right-click *Database Connections* and select *New* from the context menu, as shown in the following image.

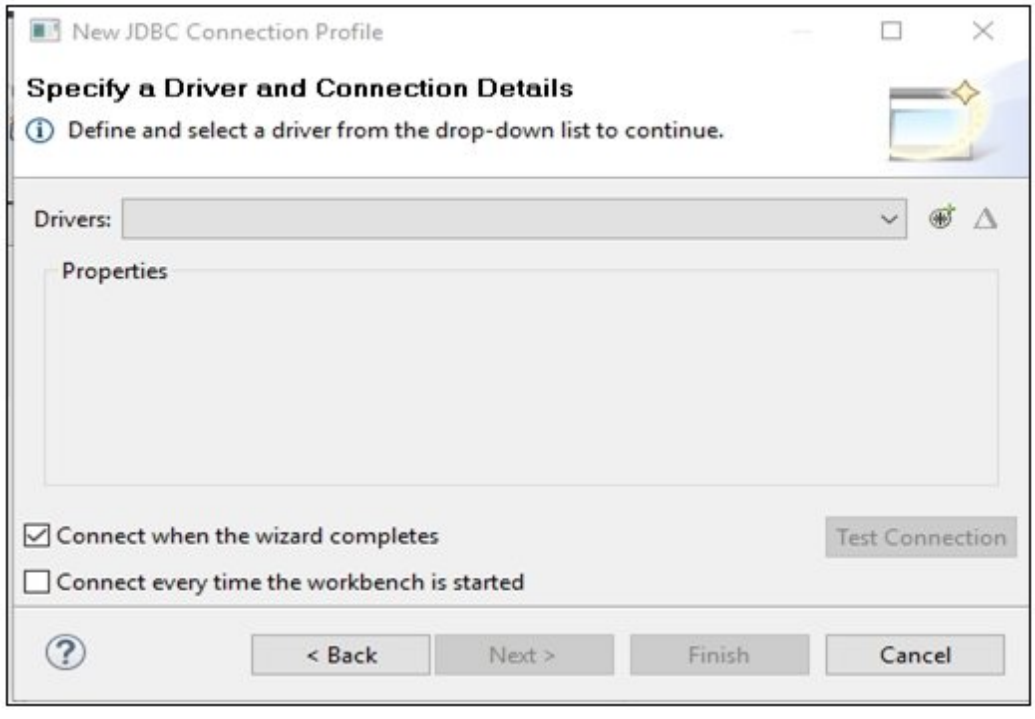


The New Connection Profile dialog opens, as shown in the following image.



Select a supported data source from the list of connection profile types. Specify a name and a description (optional) in the corresponding fields, and then click *Next*.

The New JDBC Connection Profile dialog opens, which shows the Specify a Driver and Connection Details pane, as shown in the following image.



Click the following icon to open the New Driver Definition (Specify a Driver Template and Definition Name) dialog:

Click the following icon to open an existing connection profile that you would like to edit:

Hint: If an existing data source is defined for a given connection type (for example, PostgreSQL) and a new connection profile is being created, the existing connection details are shown when the New Connection Profile dialog opens. Follow the instructions below to create a new data source template to disambiguate the connections.

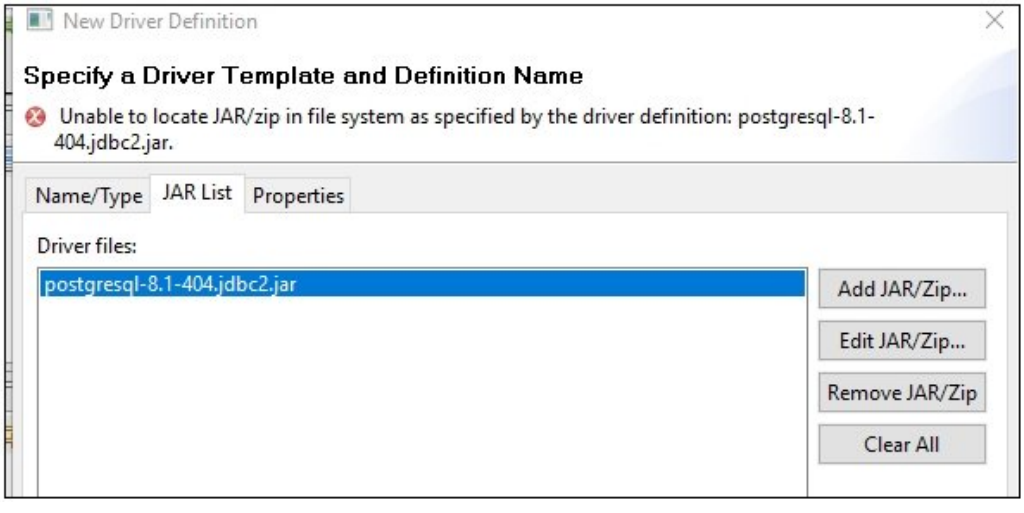
To create a new connection profile or resolve a duplicate definition:

1. Click the *New Driver Definition* icon located at the top of the New JDBC Connection Profile dialog.

   The Specify a Driver Template and Definition Name dialog opens with the Name/Type tab selected.

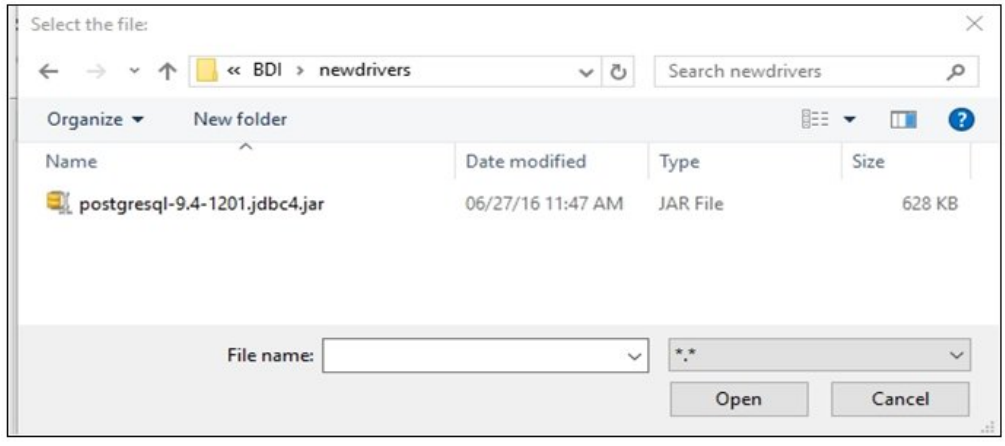
   The name of the driver, the vendor, and system version are displayed.

2. Click the driver template name, then click the JAR List tab to specify the driver for the template name, as shown in the following image.



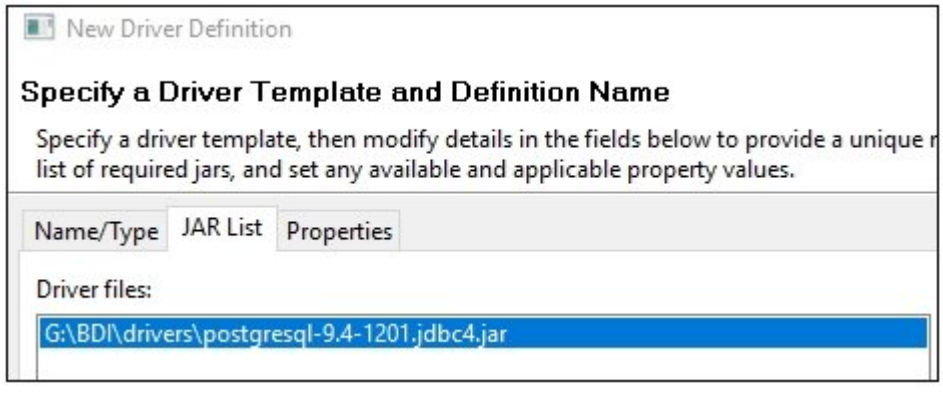
3. Remove the default driver from the template by clicking the .jar file name and then click *Remove JAR/Zip*.

4. Click *Add JAR/Zip* to locate an appropriate driver on your local file system, as shown in the following image.

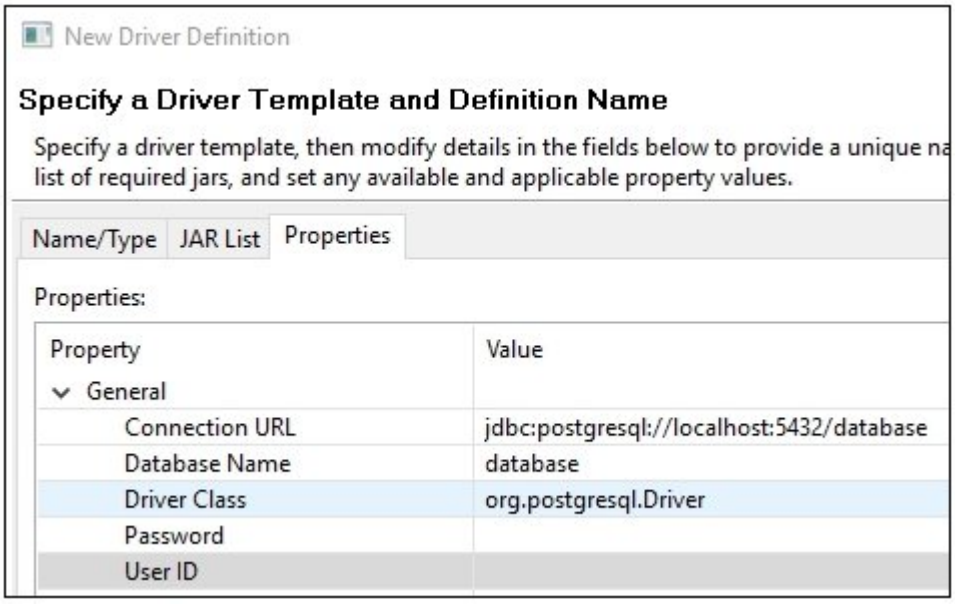


5. Select the driver file and click *Open*.

   The JAR List tab is refreshed with your new driver file, as shown in the following image.



6. Click the *Properties* tab and specify the connection information, as shown in the following image.



7. When you have finished specifying the connection information, click *OK* to close the New Driver Definition (Specify a Driver Template and Definition Name) dialog.

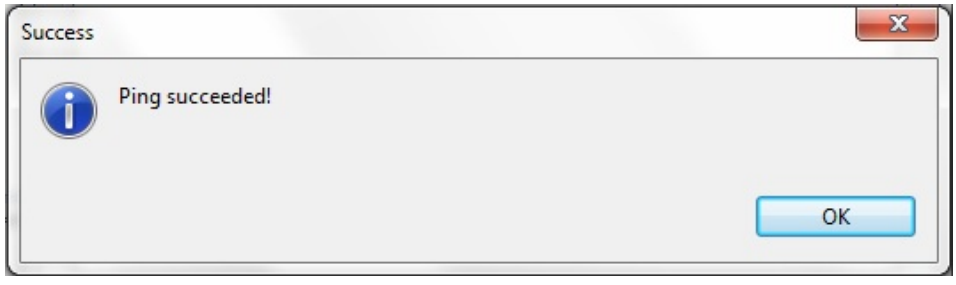
   The New Connection Profile (Specify a Driver and Connection Details) dialog is now populated with the credentials you entered in the New Driver Definition (Specify a Driver Template and Definition Name) dialog.

8. Select the *Save password* check box to save the password for the data source logon credentials.

   Hint: If a password has been entered in the New Connection Profile (Specify a Driver and Connection Details) dialog and the *Save password* check box is not selected, you will need to re-enter the password if another dialog window is opened or if the data source connection is closed and then re-opened. Otherwise, the data source will attempt to use the default password of the database driver.

9. Click *Test Connection* to verify that the connection details are valid.

If your connection configuration is valid, then the following success message is displayed.



10. Click *OK*.

    You are returned to the New Connection Profile (Specify a Driver and Connection Details) dialog.

11. Click *Finish* to save your data source connection.

Connecting to a Data Source

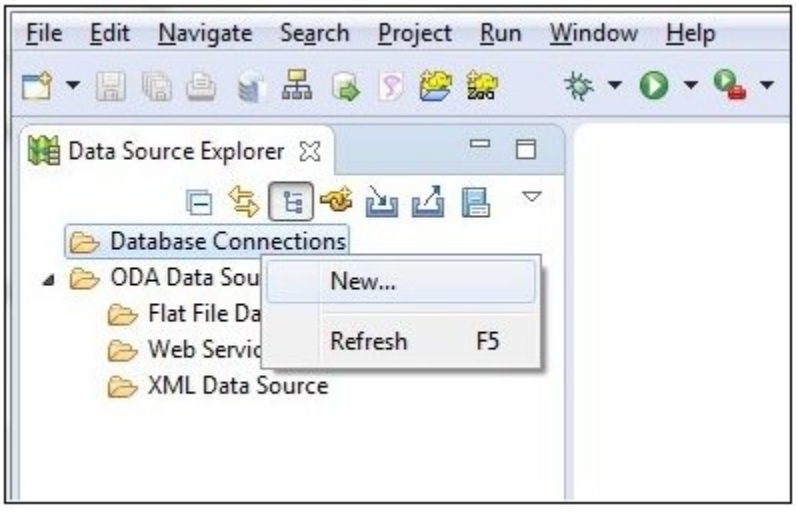
In an iWay Big Data Integrator (iBDI) project, all connections must be defined in advance in a data source profile. Start by defining the data sources in the Data Source Explorer pane.

A data source profile is a collection of properties that describe how to connect to a target and how to authorize the connection to the target. Each data source profile is unique, meaning one profile holds one set of connections. To create a second connection of the same type, such as Postgres database connections to two different Postgres servers, each server must have its own data source profile.

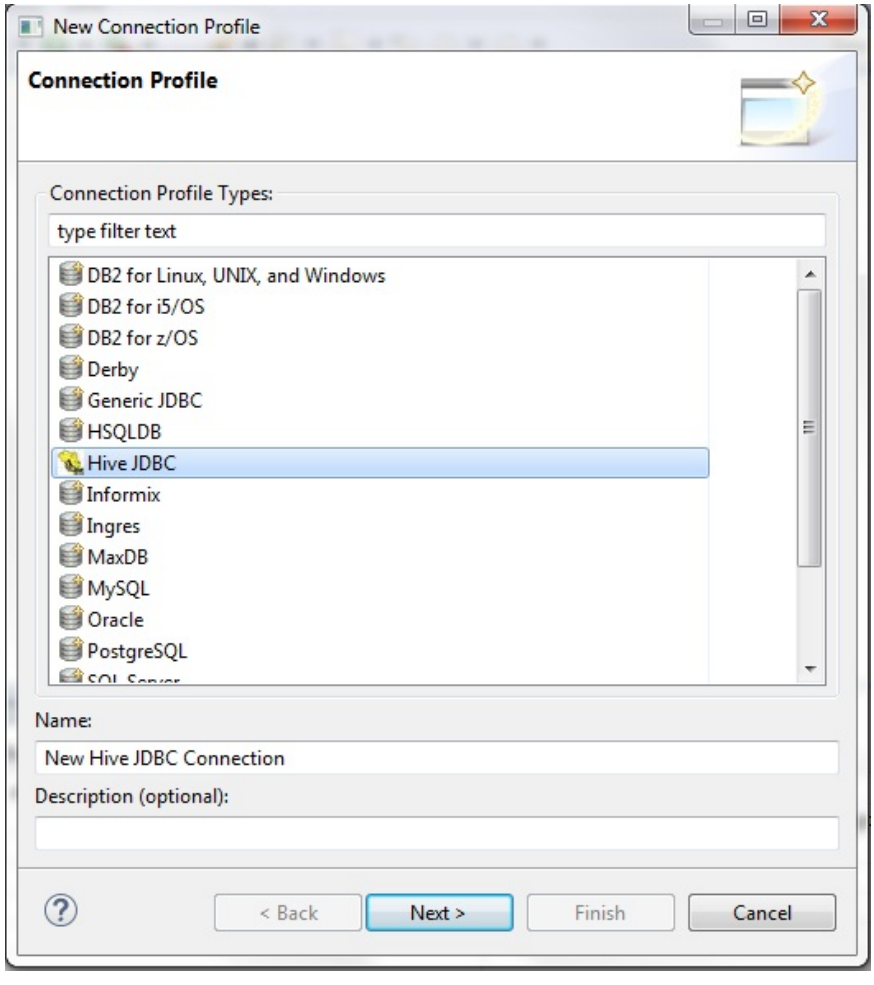
This section describes how to connect to your data source in iBDI.

Procedure: How to Create a Connection to Hive JDBC

1. In the Data Source Explorer panel, right-click *Database Connection*s and select *New* from the context menu, as shown in the following image.

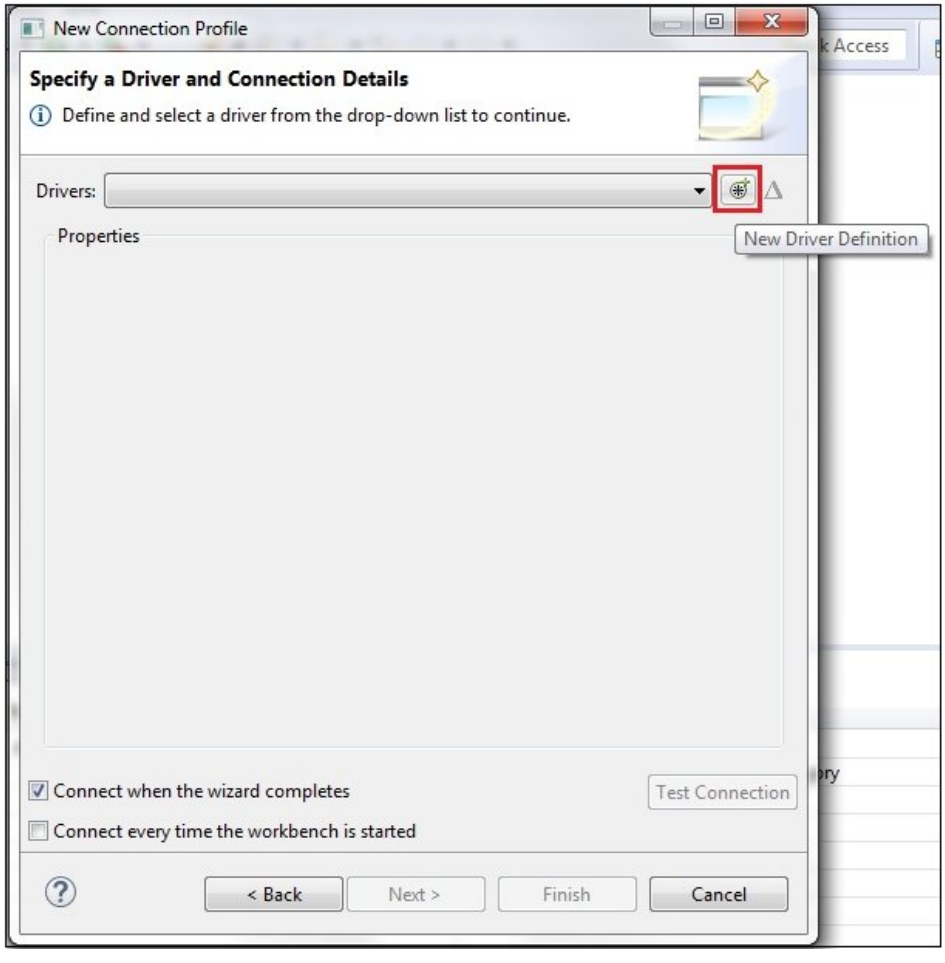


The New Connection Profile dialog opens, as shown in the following image.



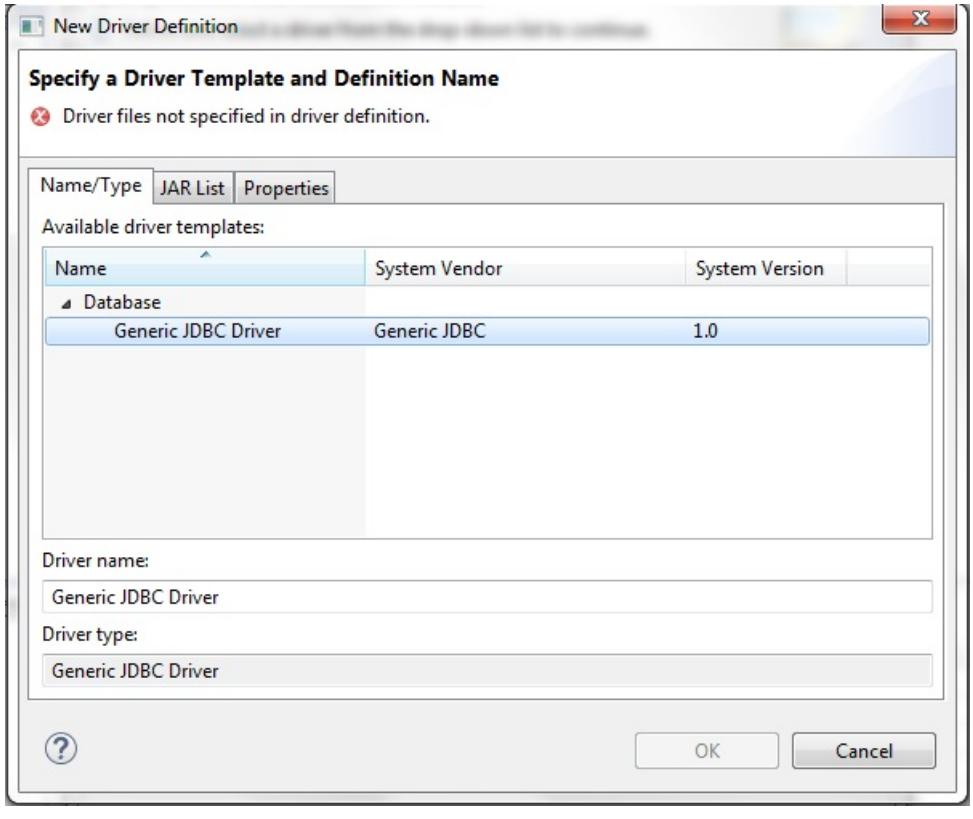
2. Select *Hive JDBC* from the list of connection profile types.
3. Specify a name (for example, *New Hive JDBC Connection*) and a description (optional) in the corresponding fields.
4. Click *Next*.

The Specify a Driver and Connection Details pane opens, as shown in the following image.



5. Click the *New Driver Definition* icon to the right of the Drivers field.

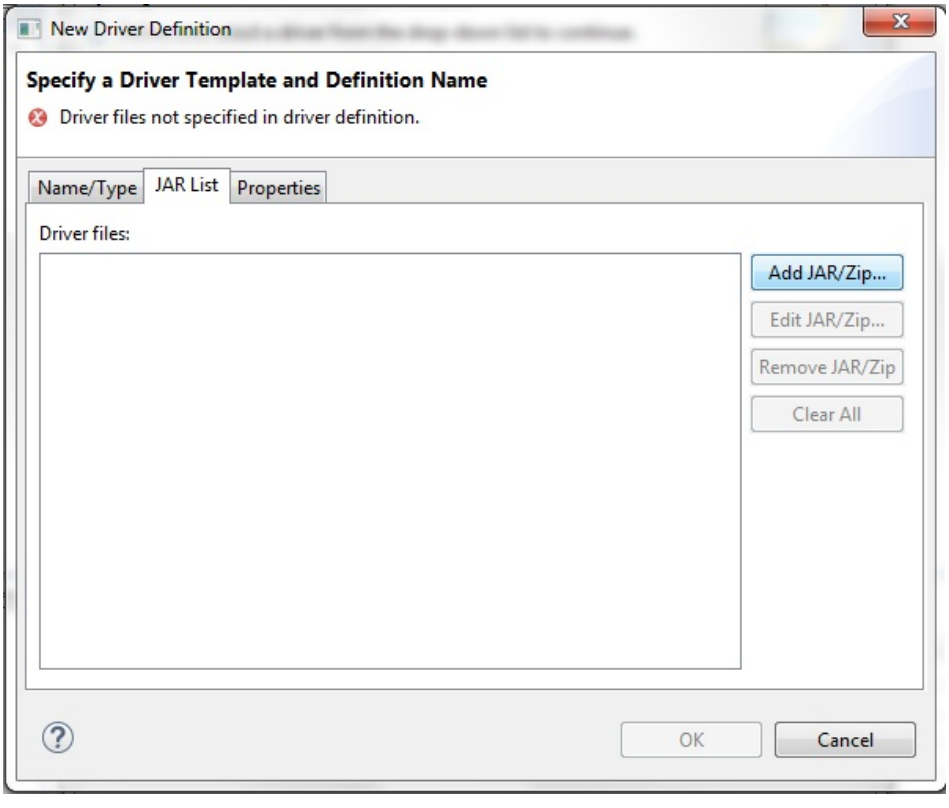
The New Driver Definition dialog opens, as shown in the following image.



6. In the Name/Type field, which is selected by default, click *Generic JDBC Driver*.

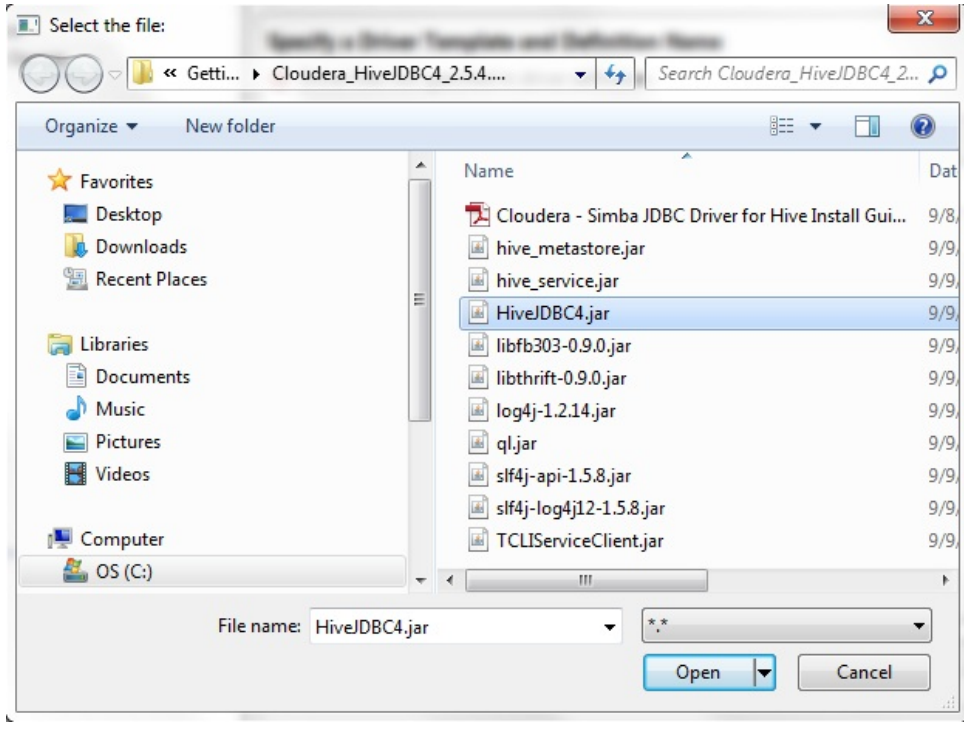
   You can change the name of the driver in the Driver name field.

7. Click the *JAR List* tab, as shown in the following image.



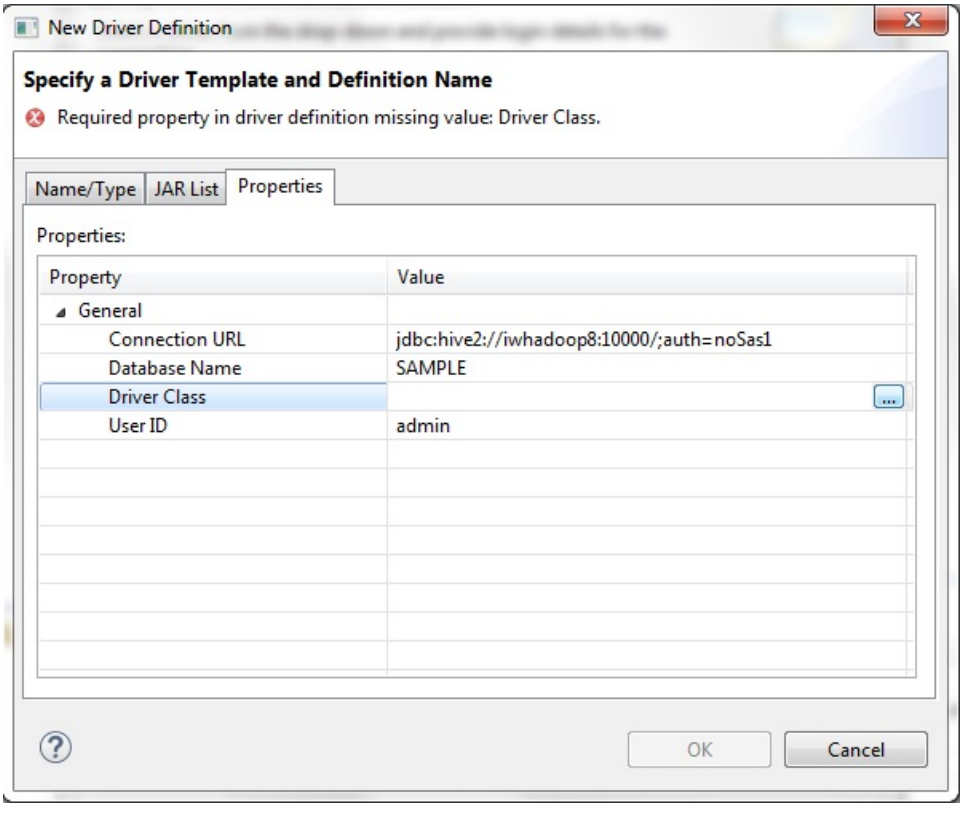
8. Click *Add JAR/Zip*.

The Select the file dialog opens, as shown in the following image.



9. Browse your file system, select the required .jar file to access your Hive JDBC (for example, *HiveJDBC4.jar*), and click *Open*.

10. Click the *Properties* tab, as shown in the following image.



11. Perform the following steps:

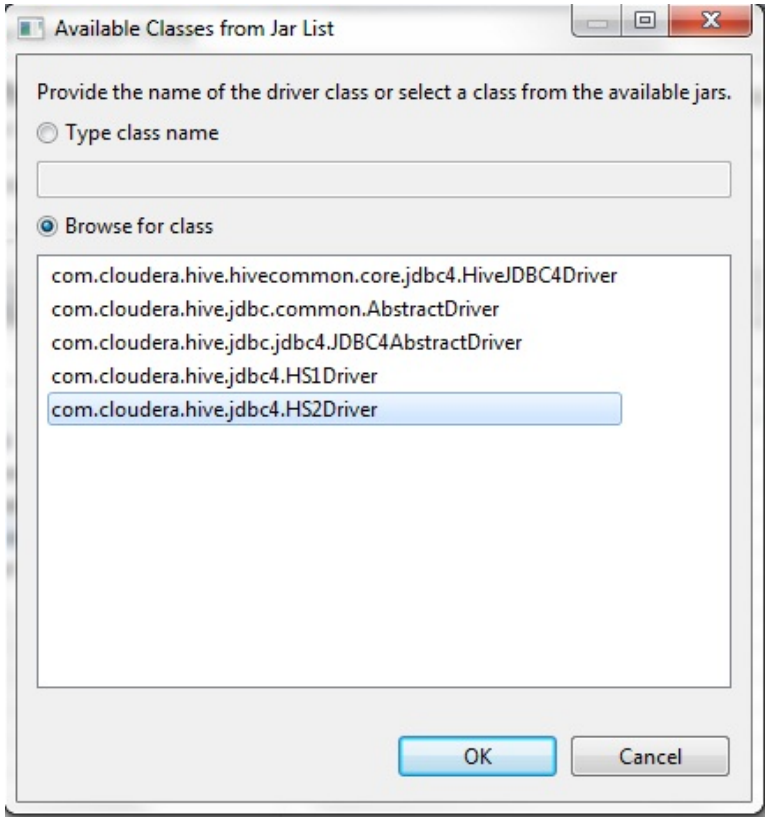
    a. Specify a connection URL for Hive.

    b. Specify the name of a database in your Hive instance that you want to explore.

    c. Specify a valid user ID.

    d. Click the *Browse* icon to the right of the Driver Class field.

The Available Classes from Jar List dialog opens, as shown in the following image.

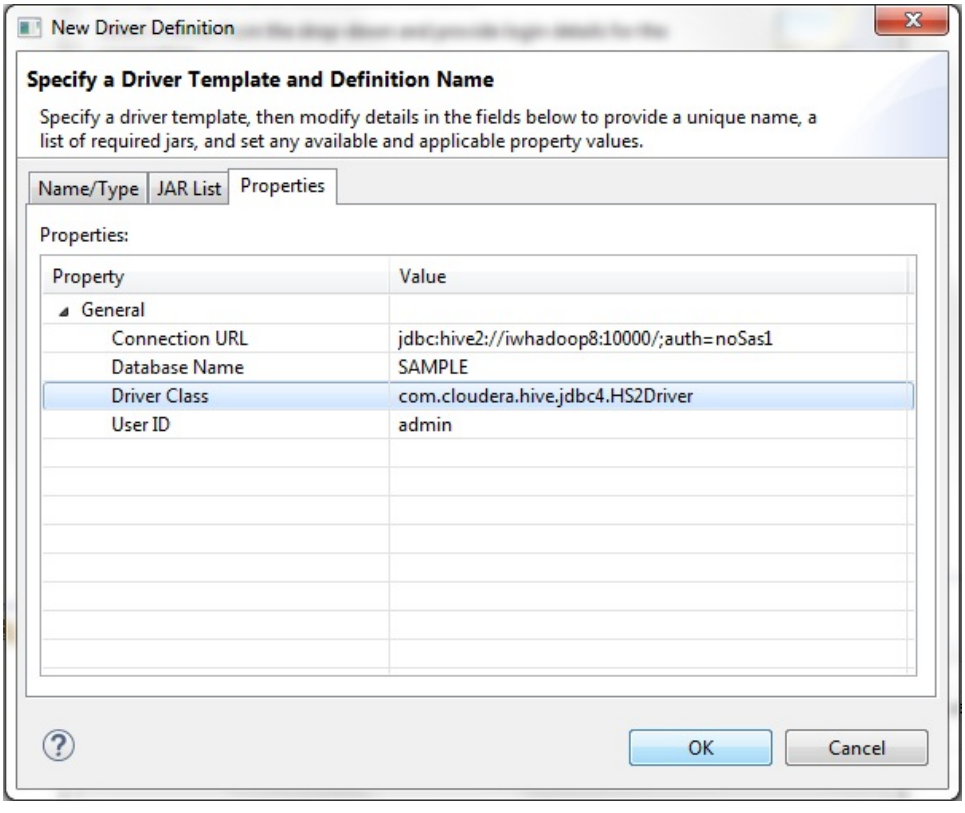


e. Click *Browse for class*, which will browse the .jar file you selected in step 9 (for example, *HiveJDBC4.jar*) for a list of classes.

f. Once a list of classes is returned, select the required class (for example, *com.cloudera.hive.jdbc4.HS2Driver*).

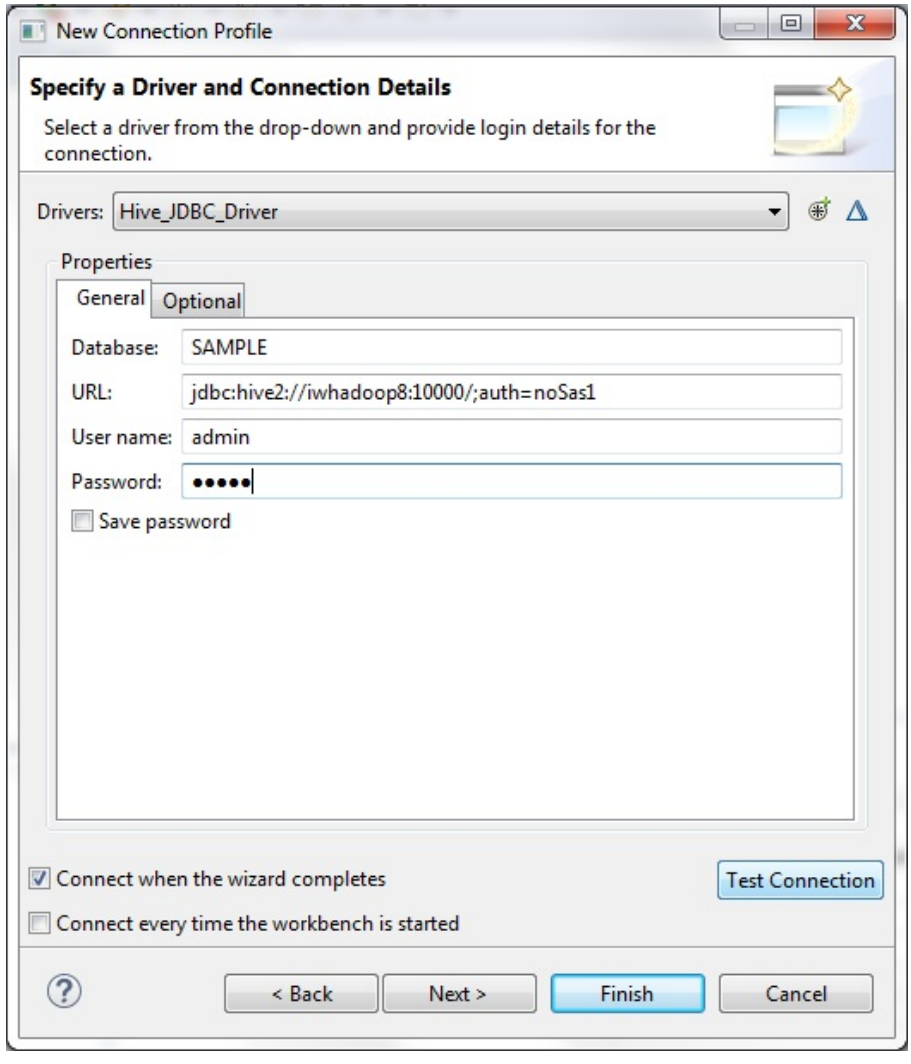
g. Click *OK*.

You are returned to the *Properties* tab, as shown in the following image.



12. Click *OK*.

The Specify a Driver and Connection Details pane opens, as shown in the following image.

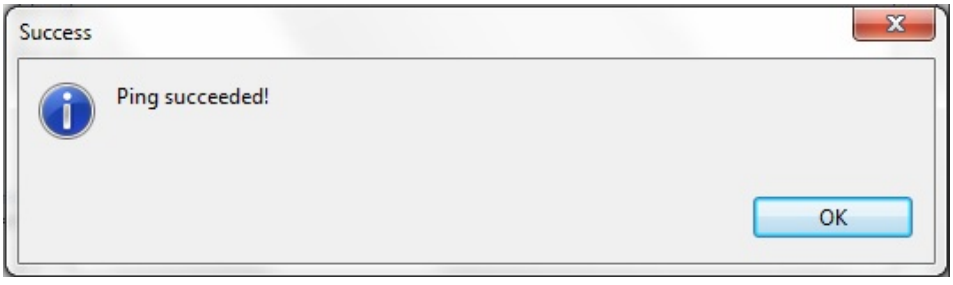


13. Specify a valid password.

    Note: You can choose to select *Save password* if you do not want to be prompted each time you connect to Hive JDBC in iBDI.

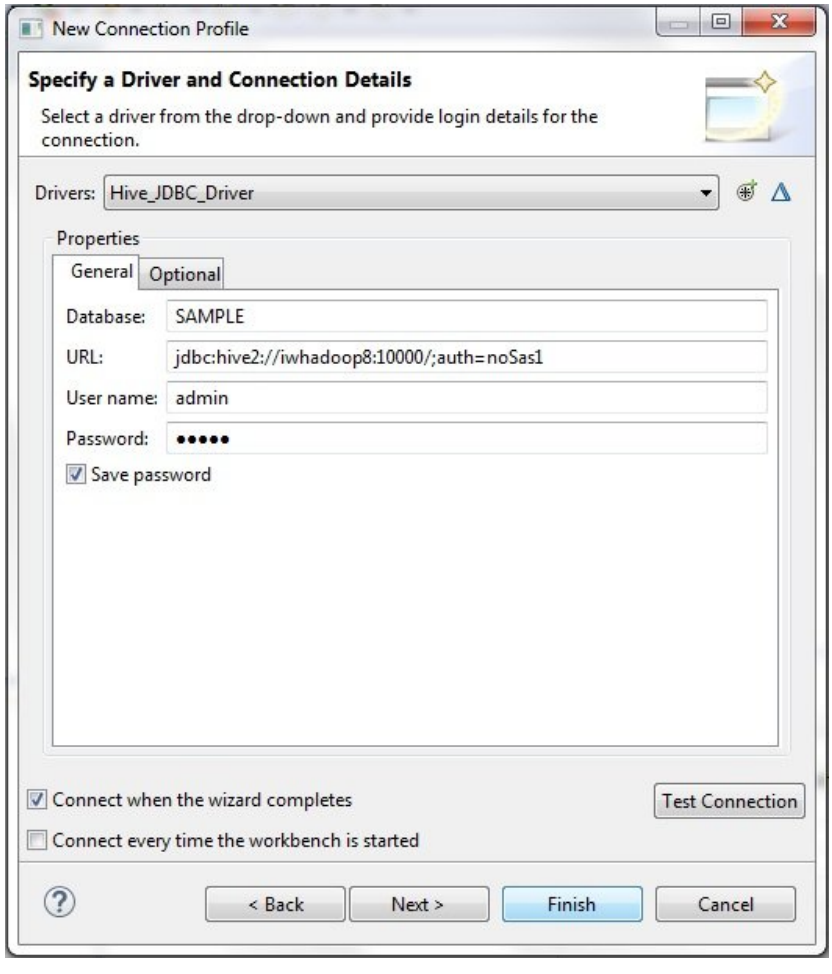
14. Click *Test Connection*.

If your Hive JDBC connection configuration is valid, then the following message is displayed.



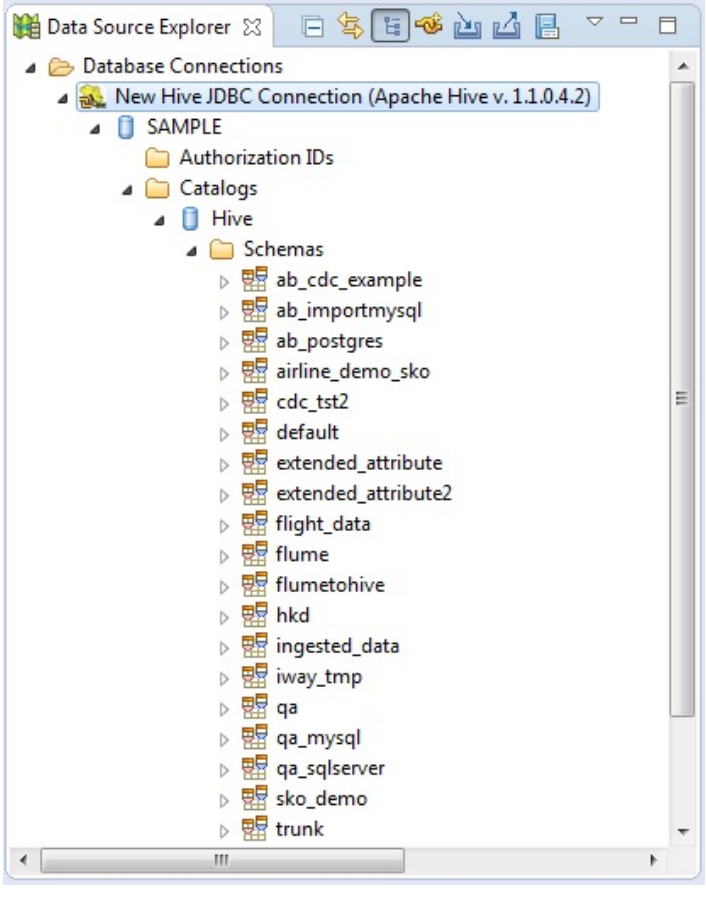
15. Click *OK*.

You are returned to the Specify a Driver and Connection Details pane opens, as shown in the following image.



16. Click *Finish*.

Your new connection to Hive JDBC is listed below the Database Connections node in the Data Source Explorer panel, as shown in the following image.

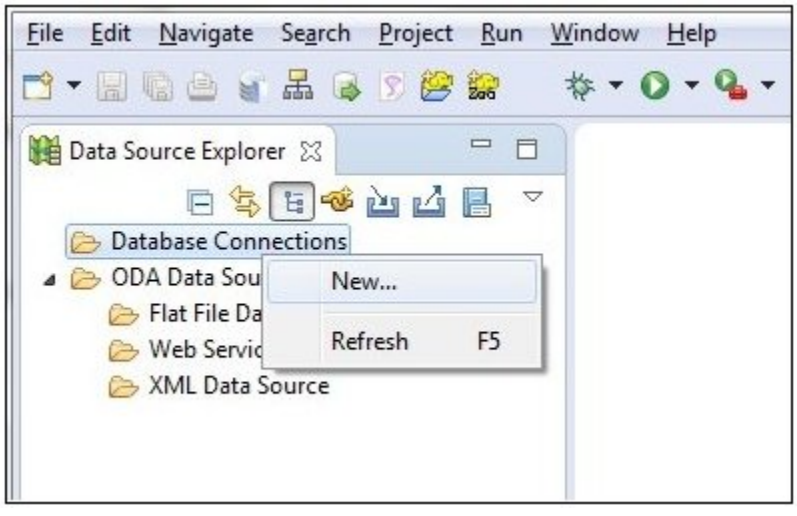


You can explore your Hive JDBC connection by expanding all of the nodes.

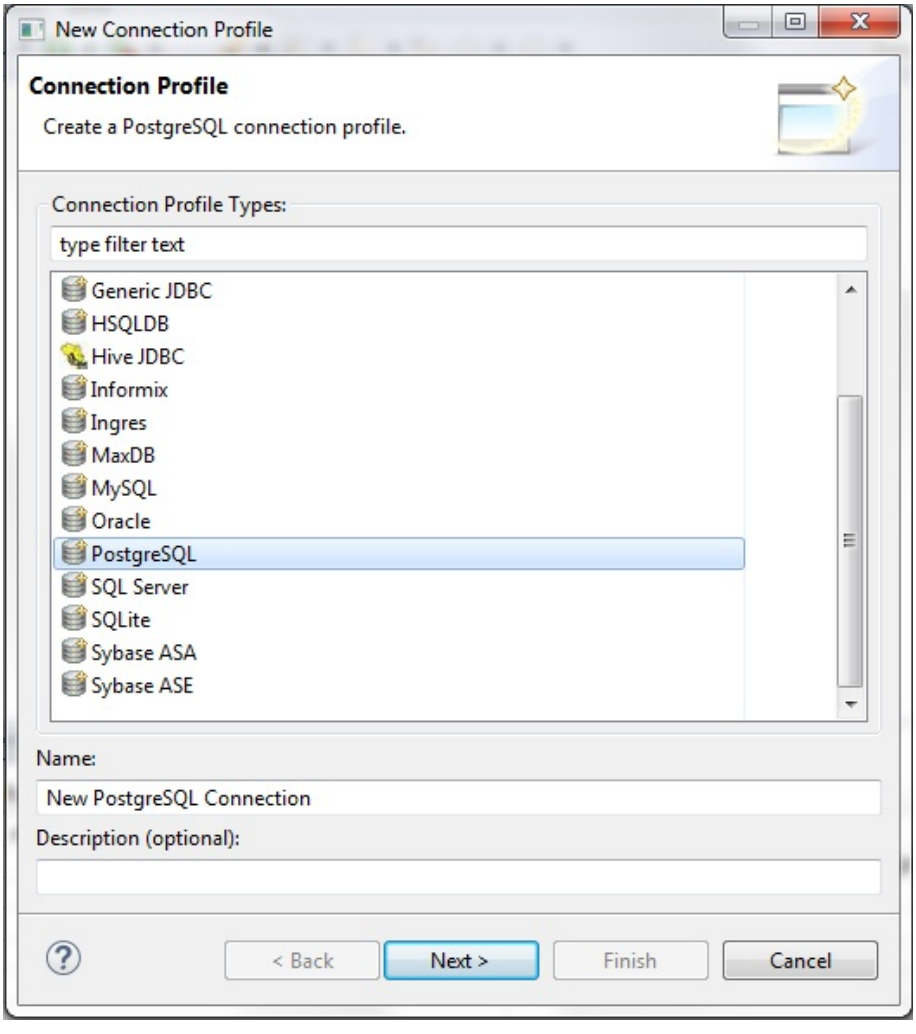
Note: JDBC URL formats for Kerberos-enabled clusters differ from non-secured environments. For more information, contact your Apache Hadoop administrator.

Procedure: How to Create a Connection to PostgreSQL

1. In the Data Source Explorer panel, right-click *Database Connections* and select *New* from the context menu, as shown in the following image.

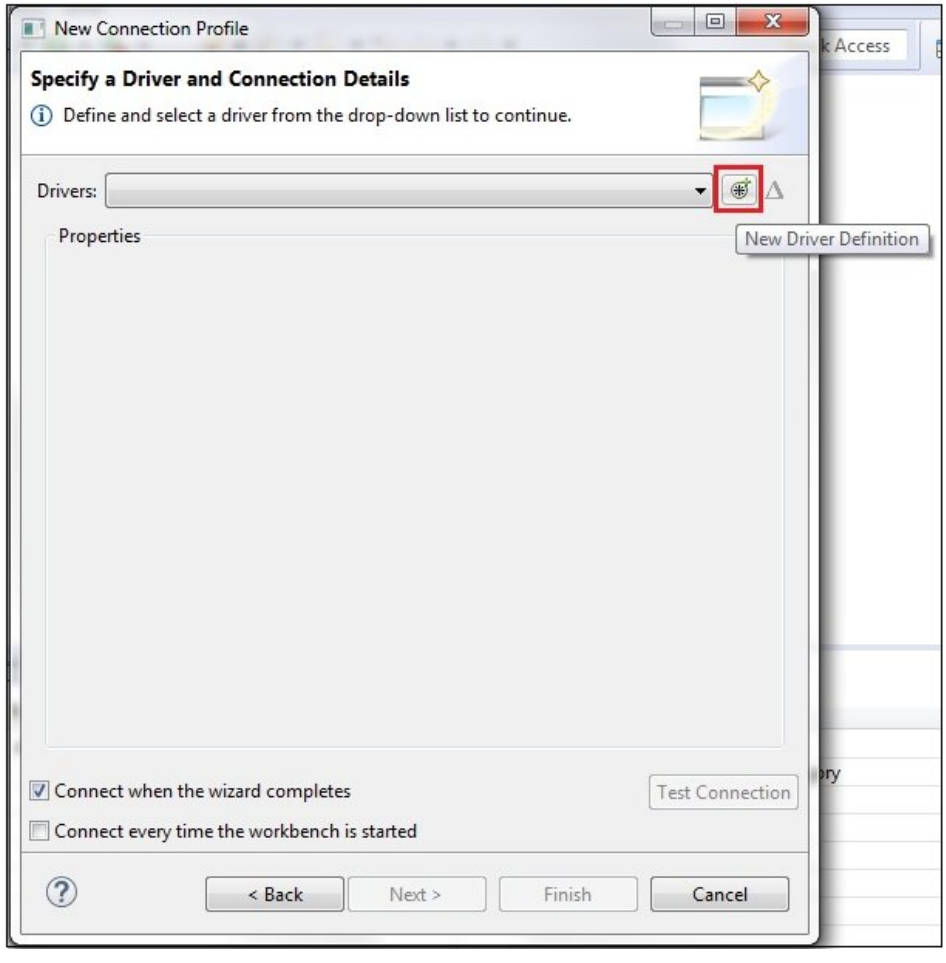


The New Connection Profile dialog opens, as shown in the following image.



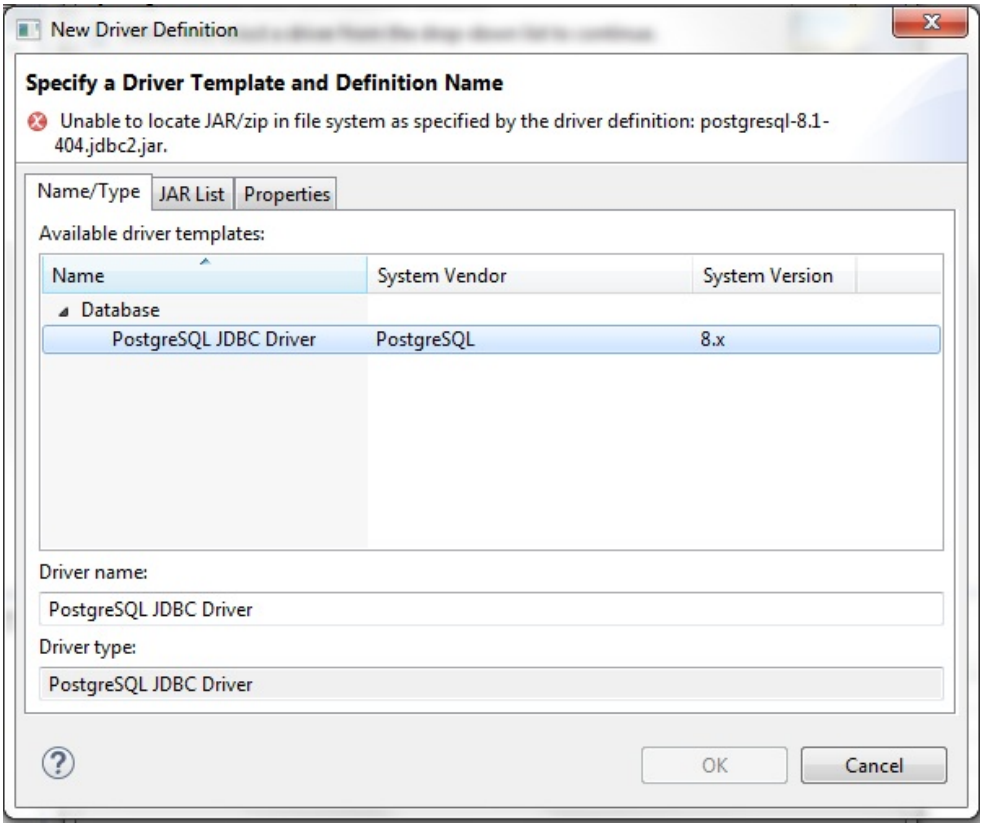
2. Select *PostgreSQL* from the list of connection profile types.
3. Specify a name (for example, *New PostgreSQL Connection*) and a description (optional) in the corresponding fields.
4. Click *Next*.

The Specify a Driver and Connection Details pane opens, as shown in the following image.



5. Click the *New Driver Definition* icon to the right of the Drivers field.

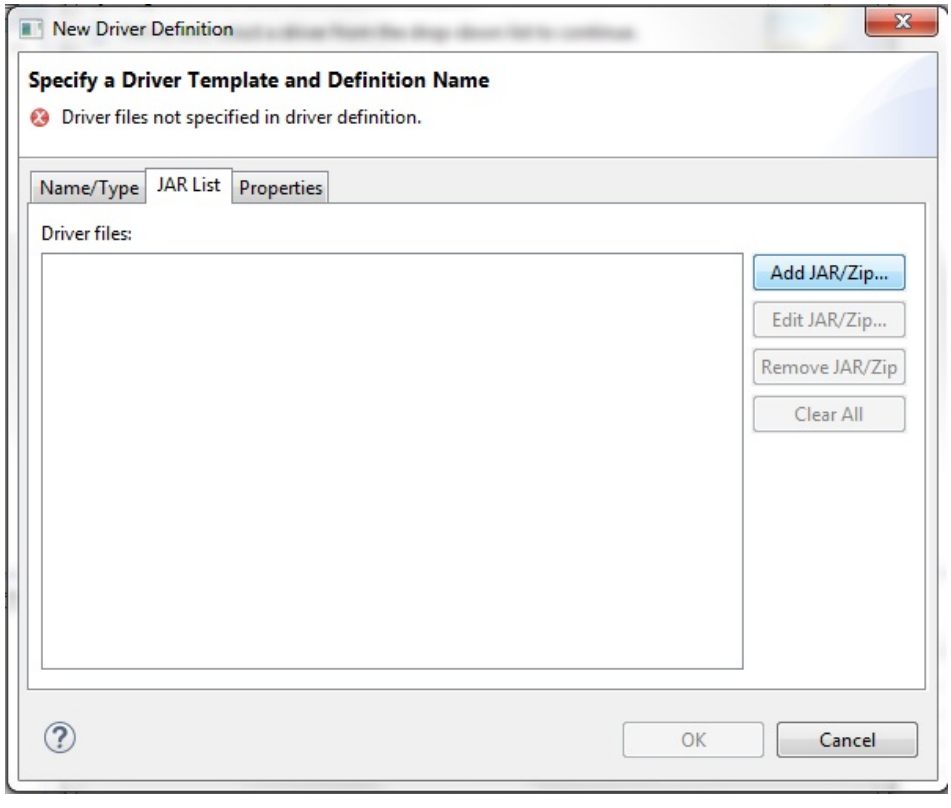
The New Driver Definition dialog opens, as shown in the following image.



6. In the Name/Type field, which is selected by default, click *PostgreSQL JDBC Driver*.

   You can change the name of the driver in the Driver name field.

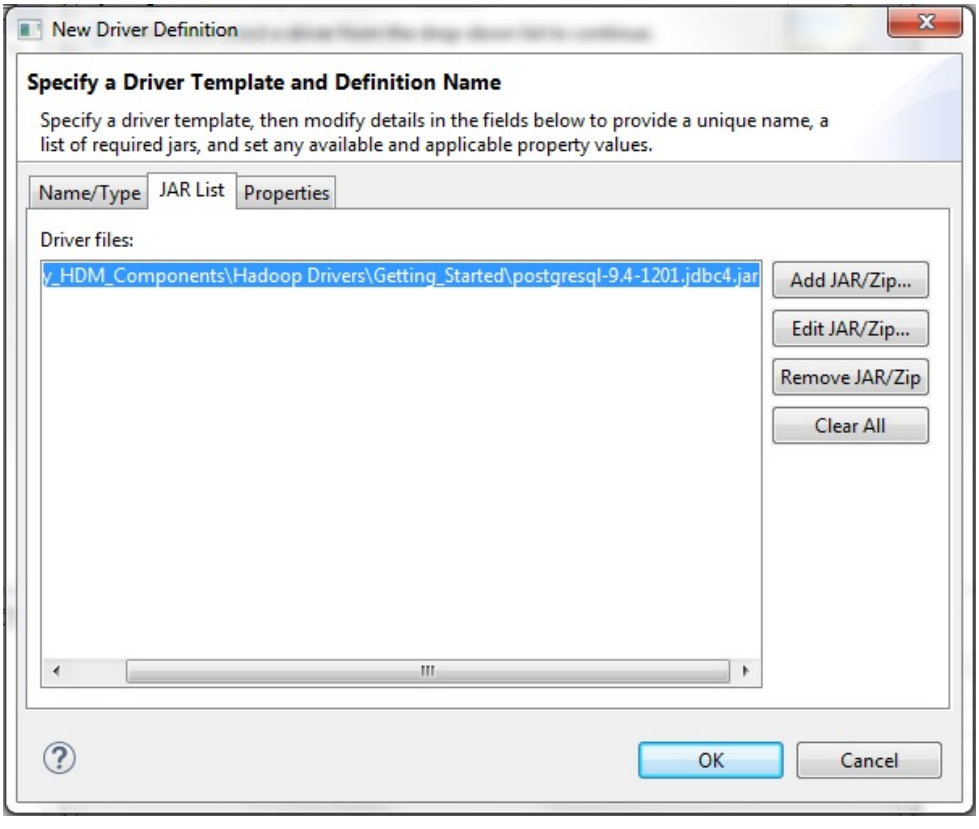
7. Click the *JAR List* tab, as shown in the following image.



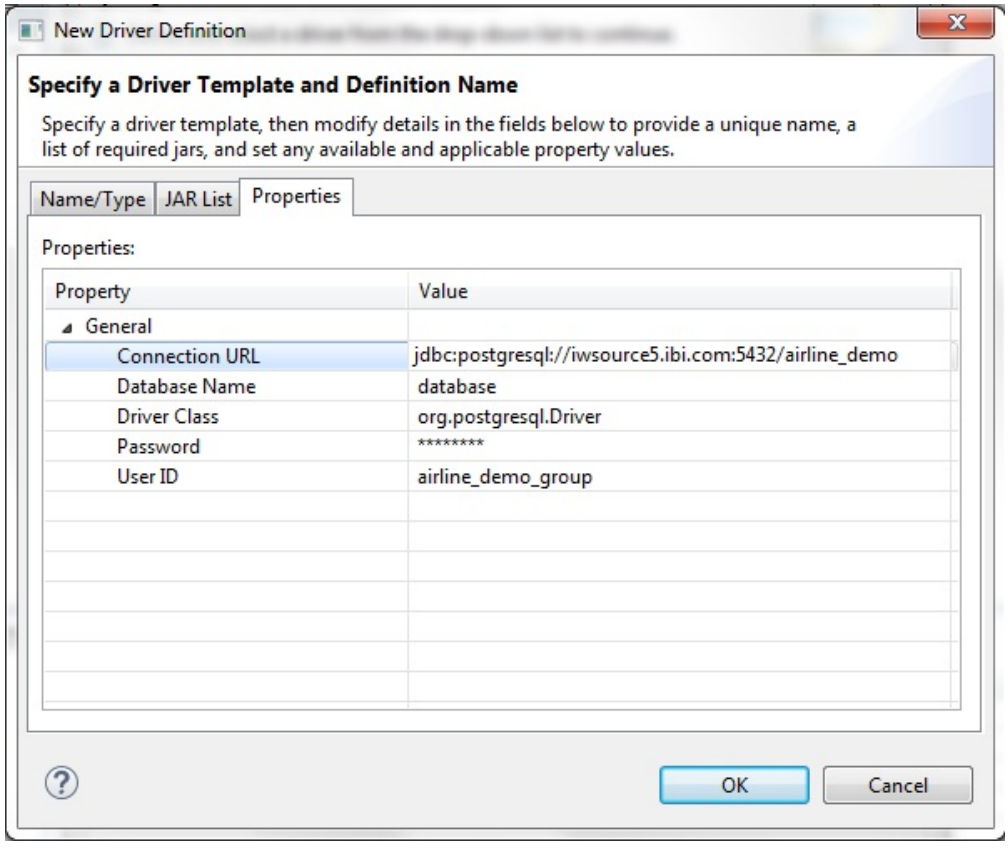
8. Click *Add JAR/Zip*.

   The Select the file dialog opens.

9. Browse your file system, select the required .jar file to access your PostgreSQL instance (for example, *postgresql-9.4-1201.jdbc4.jar*), and click *Open*.



10. Click the *Properties* tab, as shown in the following image.



11. Perform the following steps:

    a. Specify a connection URL for PostgreSQL.

    b. Specify the name of a database in your PostgreSQL instance that you want to explore.

    c. Specify a valid user ID.

    d. Specify a valid password.

    e. Click the *Browse* icon to the right of the Driver Class field.

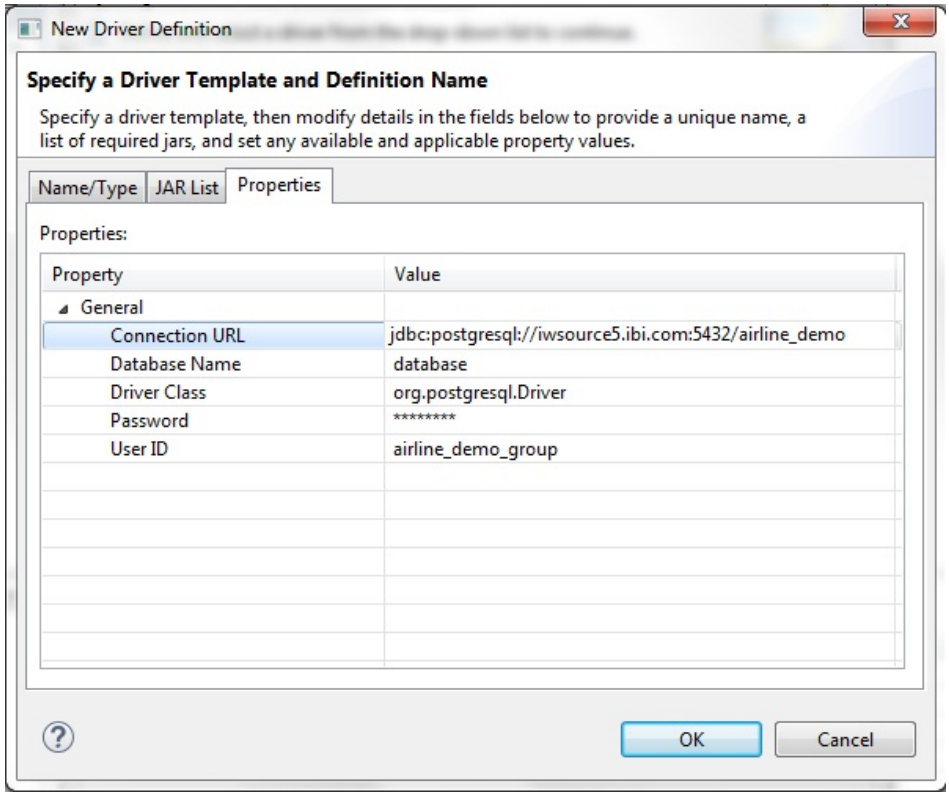
       The Available Classes from Jar List dialog opens, as shown in the following image.

    f. Click *Browse for class*, which will browse the .jar file you selected in step 9 (for example, *postgresql-9.4-1201.jdbc4.jar*) for a list of classes.

    g. Once a list of classes is returned, select the required class (for example, *org.postgresql.Driver*).

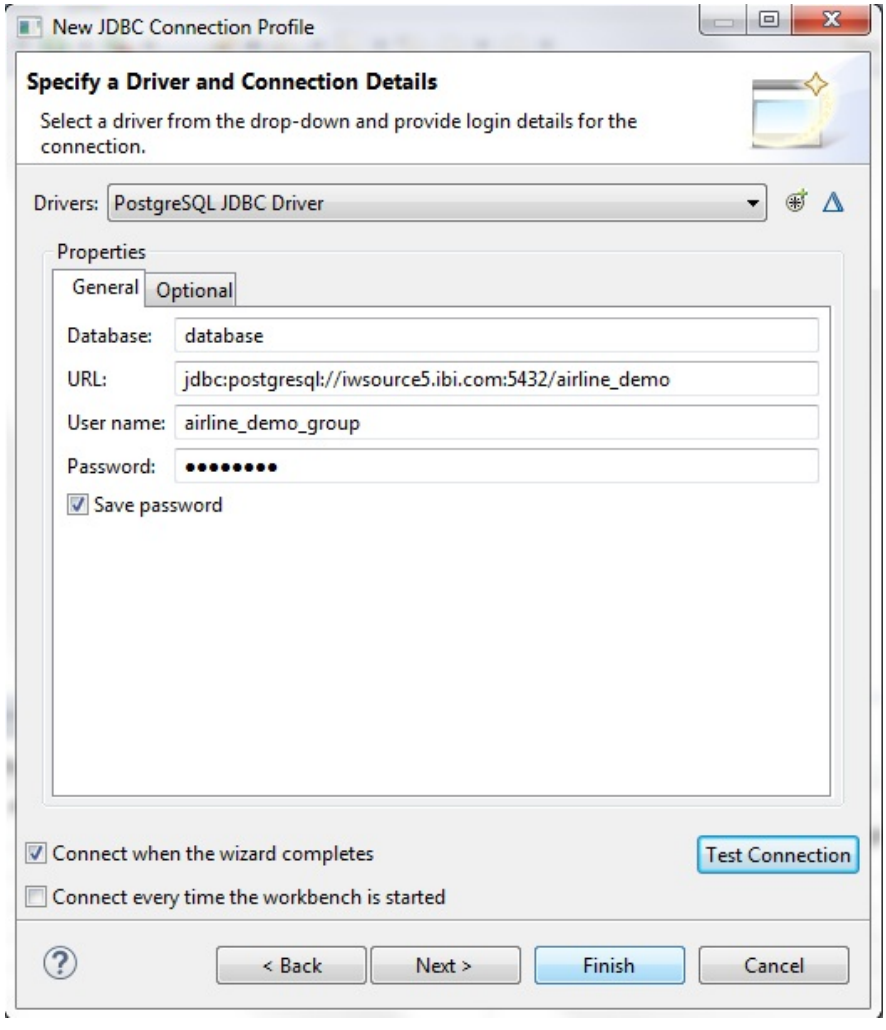
h. Click *OK*.

You are returned to the *Properties* tab, as shown in the following image.



12. Click *OK*.

The Specify a Driver and Connection Details pane opens, as shown in the following image.

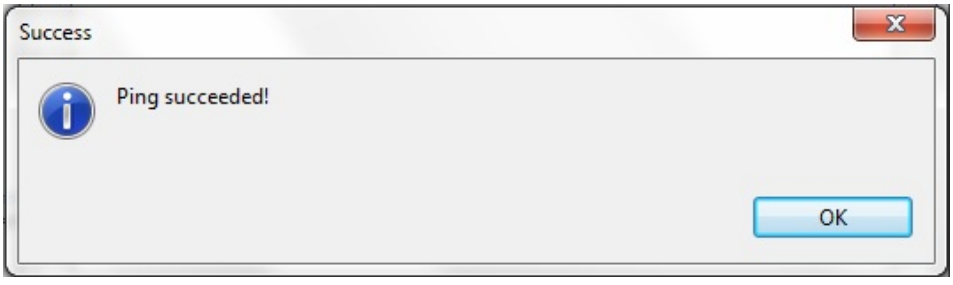


13. Specify a valid password.

    Note: You can choose to select *Save password* if you do not want to be prompted each time you connect to PostgreSQL in iBDI).

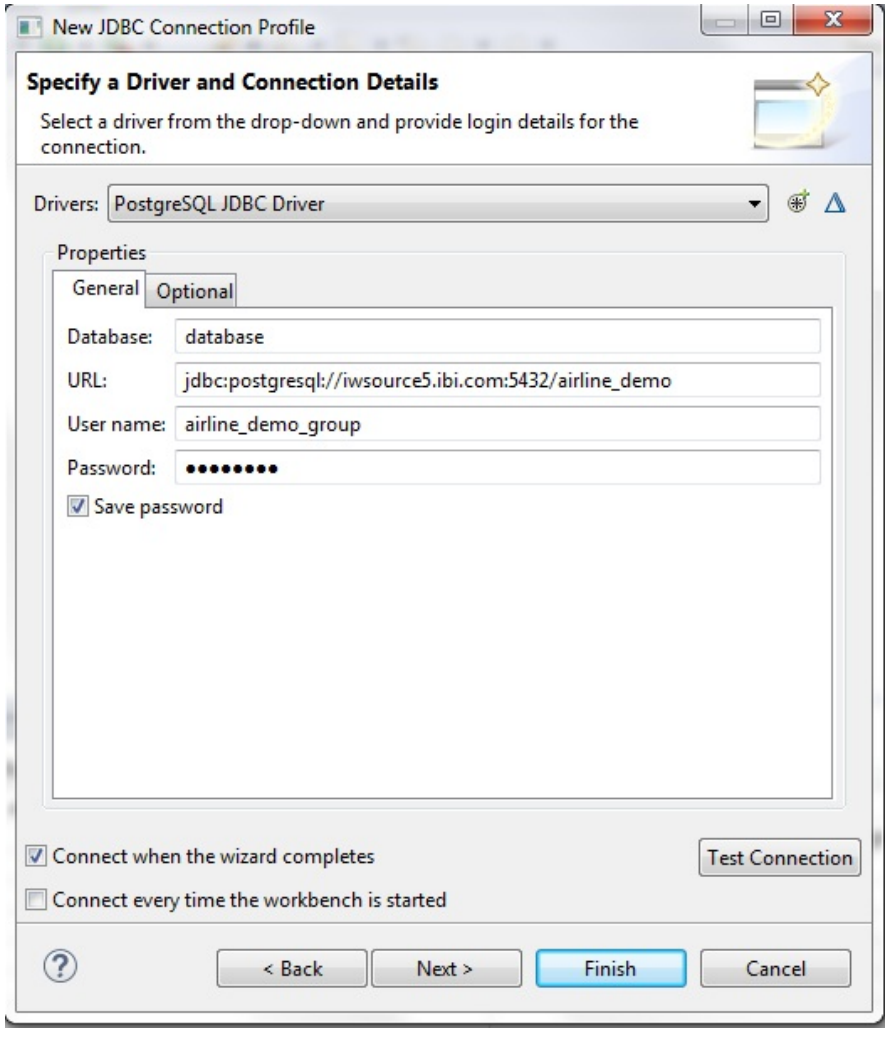
14. Click *Test Connection*.

If your PostgreSQL connection configuration is valid, then the following message is displayed.



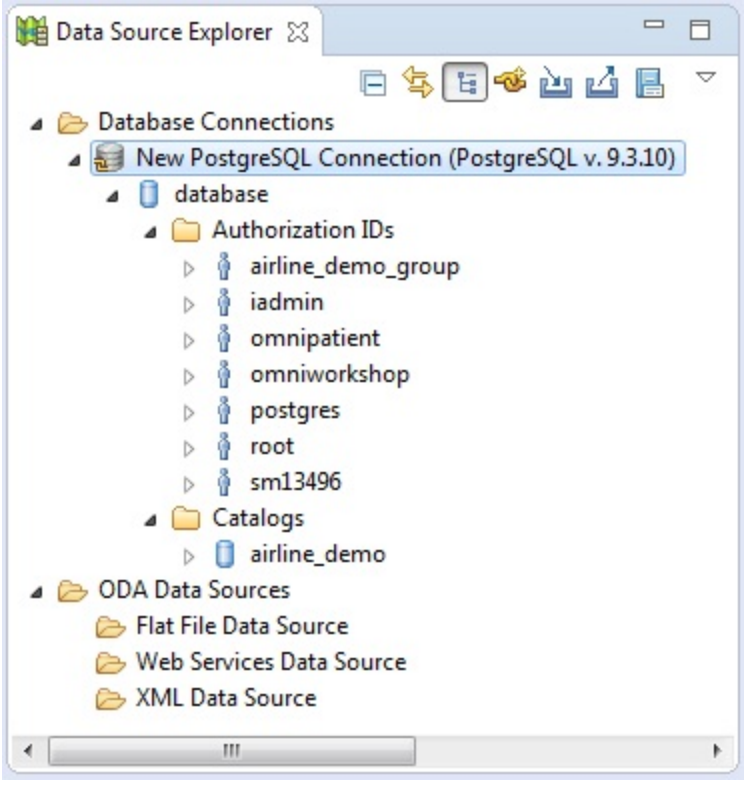
15. Click *OK*.

You are returned to the Specify a Driver and Connection Details pane opens, as shown in the following image.



16. Click *Finish*.

Your new connection to PostgreSQL is listed below the Database Connections node in the Data Source Explorer panel, as shown in the following image.



You can explore your PostgreSQL connection by expanding all of the nodes.

Creating a New Project in iBDI

A project contains all of the metadata for an iWay Big Data Integrator (iBDI) working set and all connection information for the runtime objects. A project may contain only one object or contain multiple objects depending on the requirements of the organization and design patterns. Pre-defined categories of Big Data operations appear in the project automatically as the operations are implemented, such as Pipelines or Flumes.

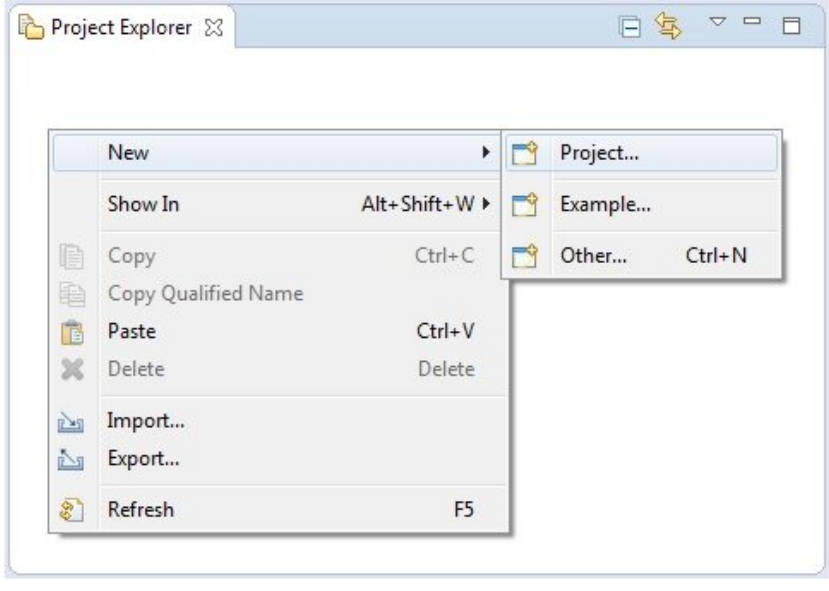
You can create projects by right-clicking anywhere within the Project Explorer pane, selecting *New* from the context menu, and then clicking *Project*. You can also click *File* from the main menu, select *New* from the context menu, and then click *Project*. When the project is created, a new entry appears in the Project Explorer pane for the project and subfolders appear for each of the iBDI operations.

Creating a pipeline opens the pipeline editor, which enables all new Spark-based functionality. Non-Spark or Hadoop operations are enabled by creating a new operation of the type from the project.

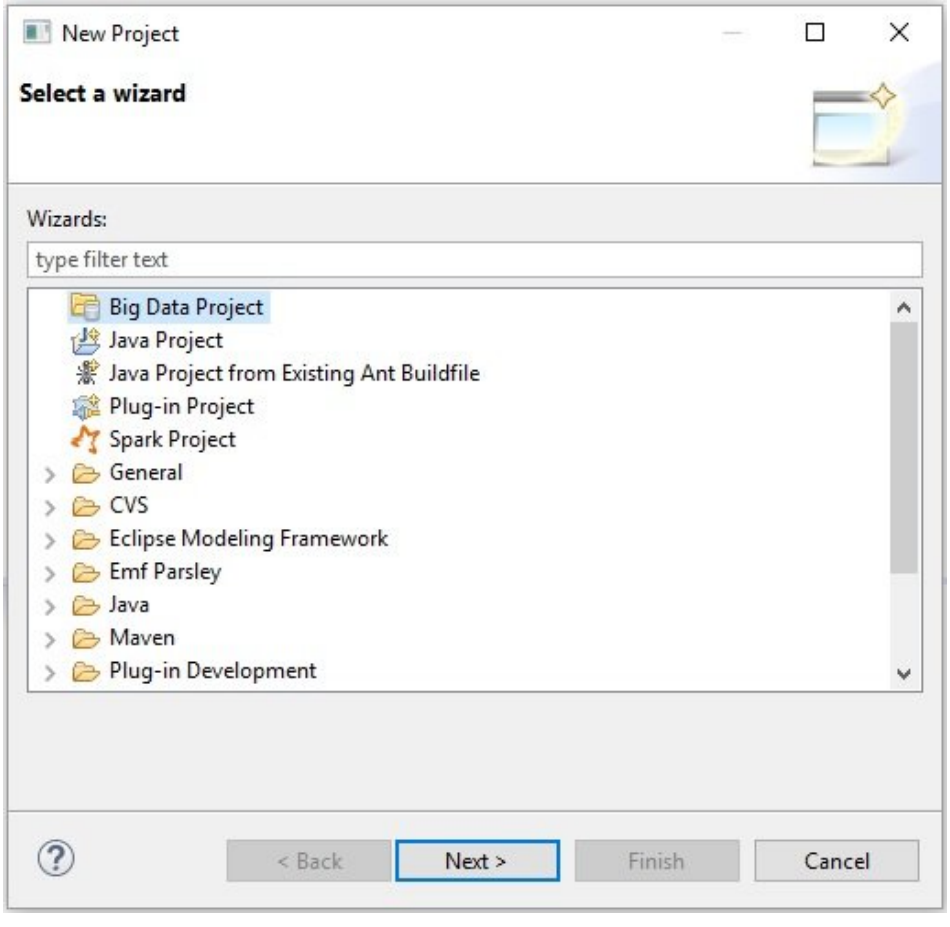
This section describes how to create a new project in iBDI.

Procedure: How to Create a New Project

1. Right-click anywhere in the Project Explorer panel, select *New* from the context menu and then click *Project*, as shown in the following image.

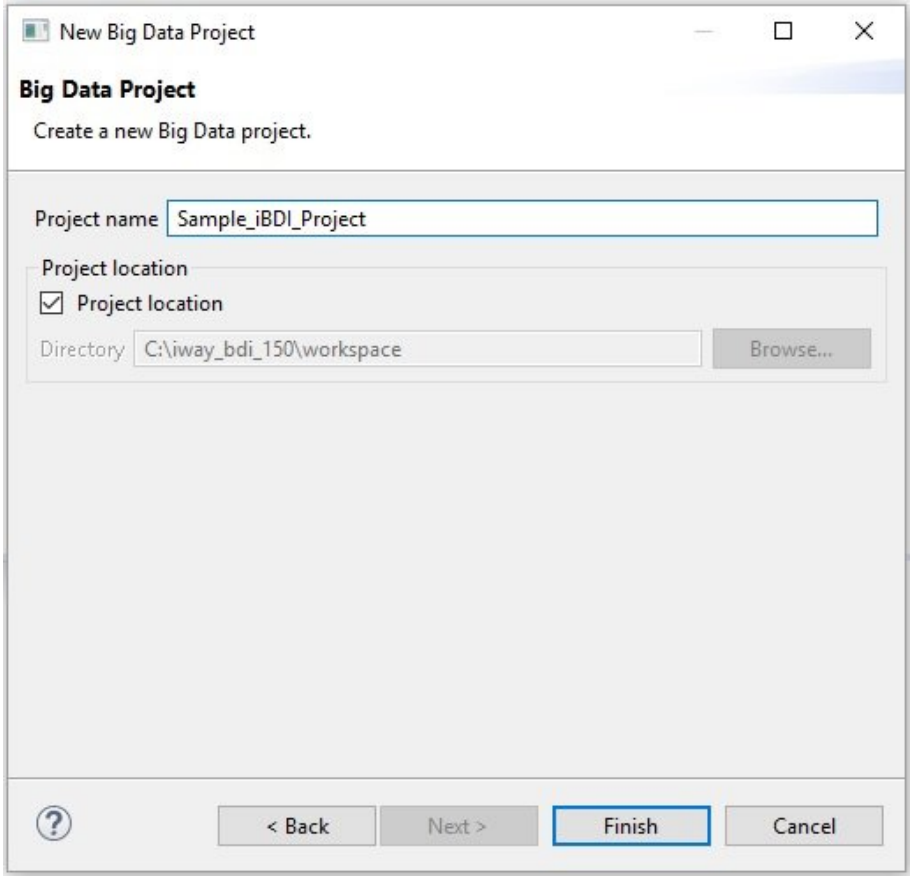


The New Project dialog opens, as shown in the following image.



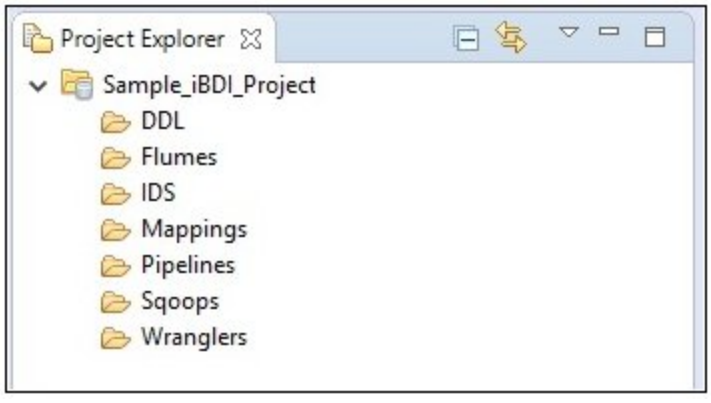
2. Select *Big Data Project* from the list of wizards and then click *Next*.

The New Big Data Project dialog opens, as shown in the following image.

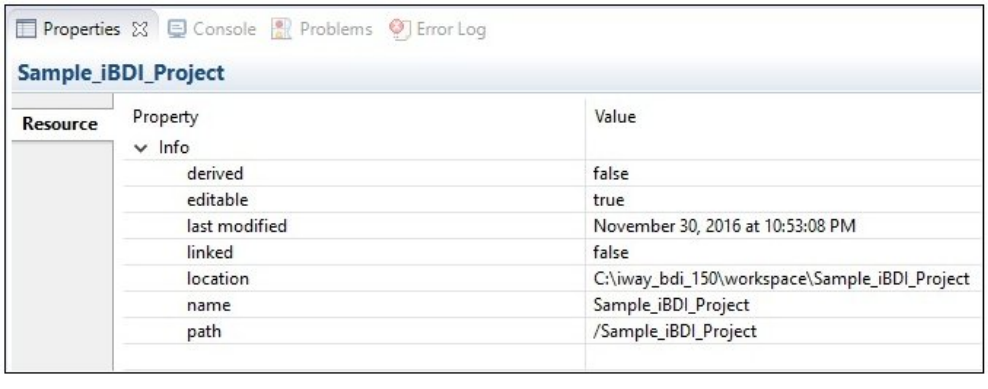


3. Specify a unique name for your new project in the Project name field.
4. Specify a location for your project or accept the default, which will save your project to the workspace directory by default.
5. Click *Finish*.

Your new project (for example, Sample_iBDI_Project) is listed in the Project Explorer panel with corresponding subfolders, as shown in the following image.

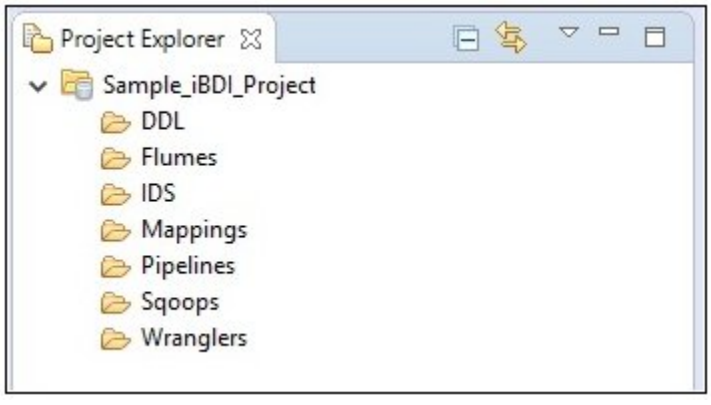


Properties for your project can be viewed in the Properties tab, as shown in the following image.



Project Components

There are various project components that are supported by iWay Big Data Integrator (iBDI) for each project. Project components are stored in the corresponding subfolders that are created for an iBDI project, as shown in the following image.



Multiple project componentsan iBDI project

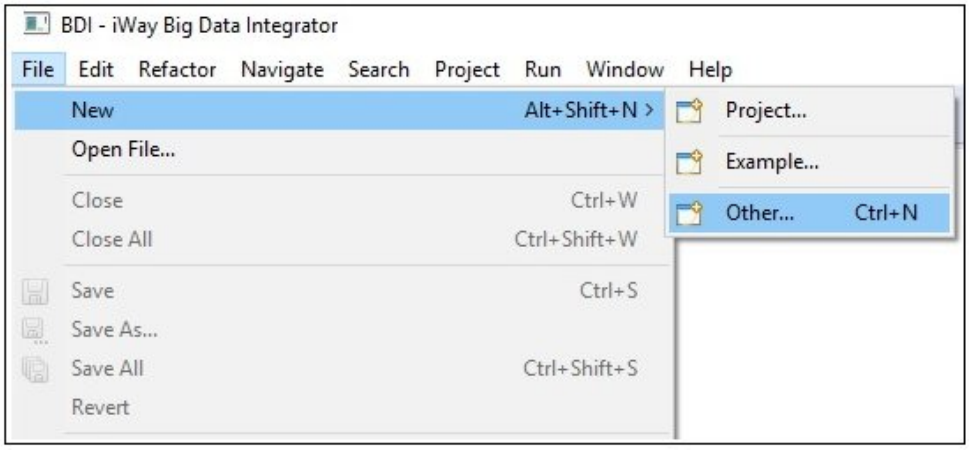
The following table lists and describes the supported project components.

Project Component / Subfolder	Description
DDL	Data Definitions. You can export a model from any RDBMS and iBDI will translate it to HQL and create the tables in Hive.
Flumes	Streaming inbound data.
IDS	Input Document Specifications, for use in Mapper table creation.
Mappings	Hive transformations.
Pipelines	Spark pipelines for multiple sources and transformations.
Sqoops	Relational inbound data.
Wranglers	Apply schemas to transform HDFS sources.

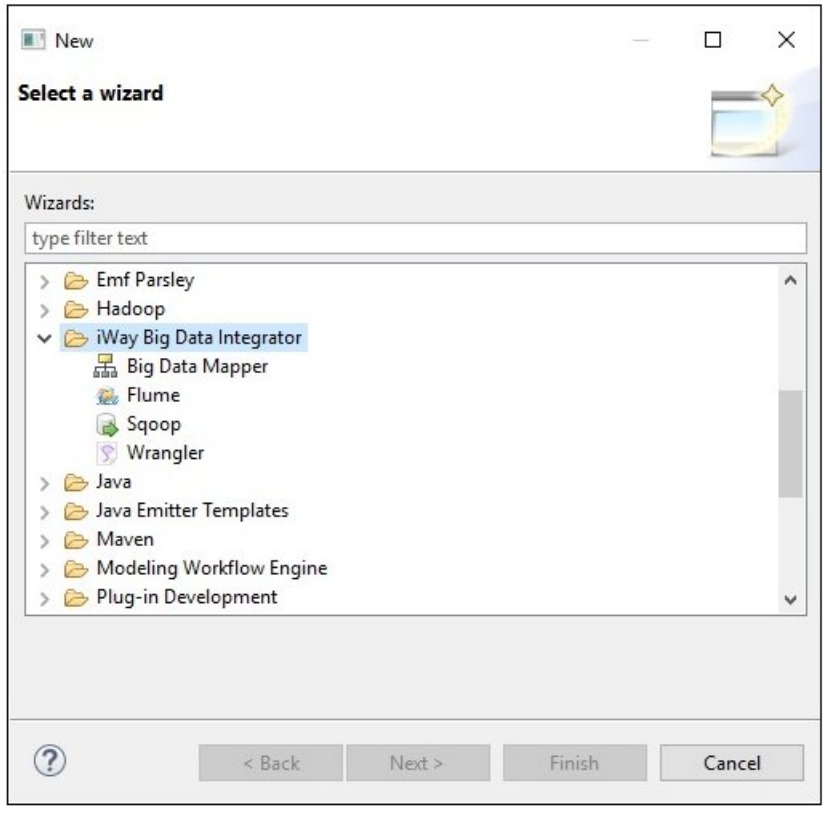
Depending on your requirements, multiple project components can be configured in a single iBDI project.

Note: Each component in an iBDI project consumes memory when this project is opened. For example, a large iBDI project containing many project components will be slower to open.

You can create a new project component by clicking *File* from the main menu, selecting *New* from the context menu, and then clicking *Other*, as shown in the following image.

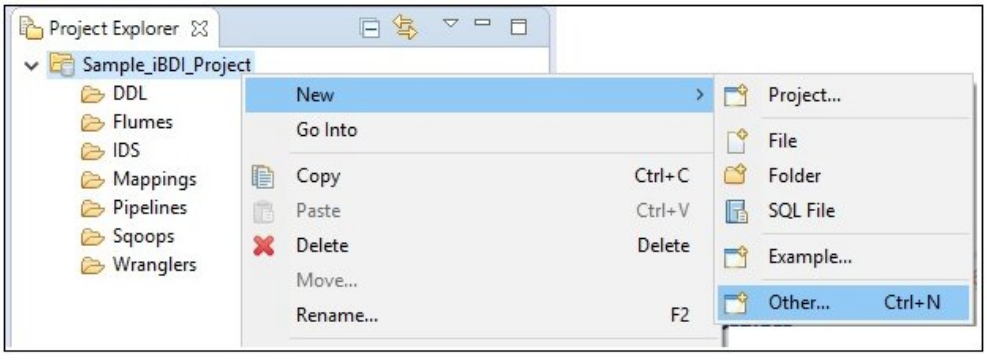


The New - Select a wizard dialog opens, as shown in the following image.

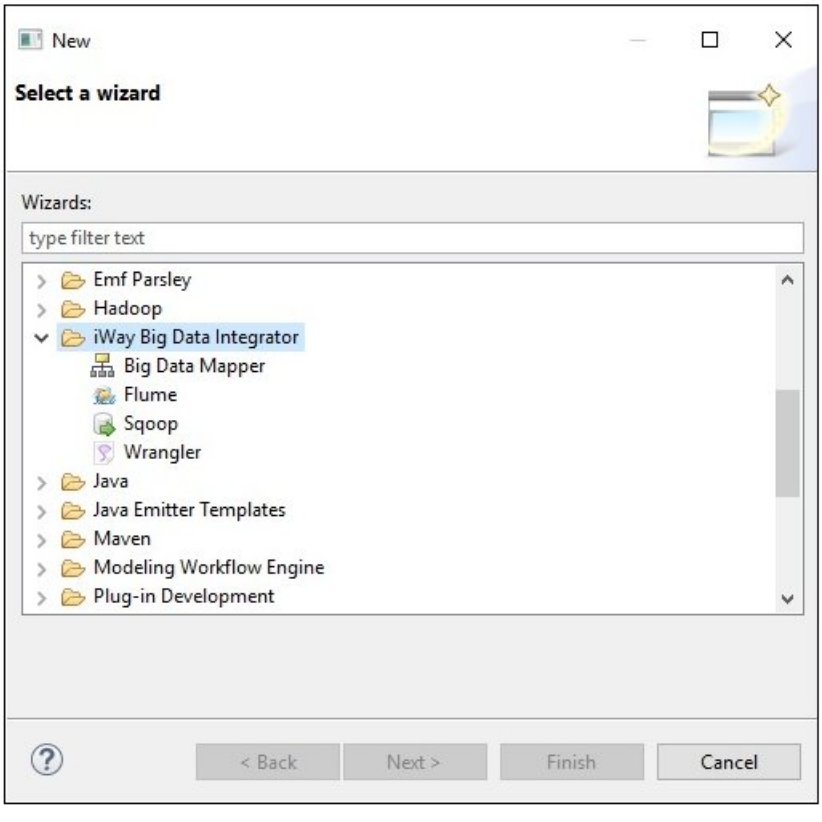


Select the type from the list or type the name in the Wizards field.

The project component can also be created directly from the iBDI project by right-clicking the project, selecting *New* from the context menu, and then clicking *Other*, as shown in the following image.



The New - Select a wizard dialog opens, as shown in the following image.



Select the type from the list or type the name in the Wizards field.

Chapter 5

Defining Data in the Data Source Explorer

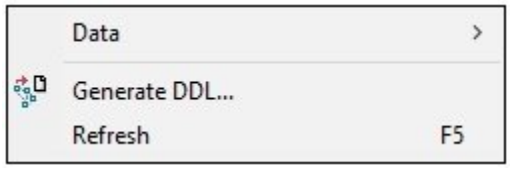
This chapter describes how to define data in the Data Source Explorer and use the Generate DDL wizard to export a model from any RDBMS for iWay Big Data Integrator (iBDI) to translate to HQL and create the tables in Hive.

In this chapter:

- Using the Data Source Explorer for Extract, Transform, and Load Operations
- Using the Generate DDL Wizard

Using the Data Source Explorer for Extract, Transform, and Load Operations

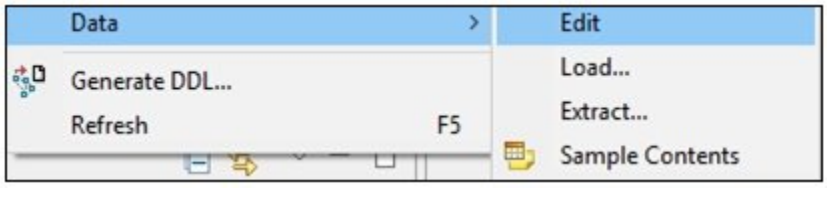
Connect to a data source in the Data Source Explorer and navigate to a schema. Open the schema to view the list of tables. Select a table by placing the cursor on the table and then right-click the table name. A context menu opens, as shown in the following image.



Select *Refresh* if there have been recent changes to the table that do not appear in table view. Select *Generate DDL* to open the Generate DDL wizard. For more information, see *Using the Generate DDL Wizard* on page 77.

Understanding the Data Submenu Options

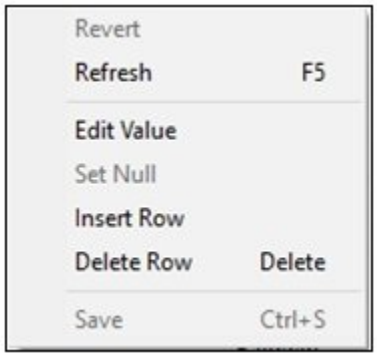
Selecting *Data* opens a submenu, as shown in the following image.



This section describe each of these submenu options.

Edit

The **Edit** submenu option opens the first 500 records of the file in editor mode. Values can be changed (depending on the permissions in the underlying databases) by selecting and then right-clicking a column. A context menu opens, as shown in the following image.



A cell (row/column) combination can be edited by clicking into that cell and changing the value. Select *Revert* if you want to remove (undo) the change that was made or *Save* to save the changed data in the database.

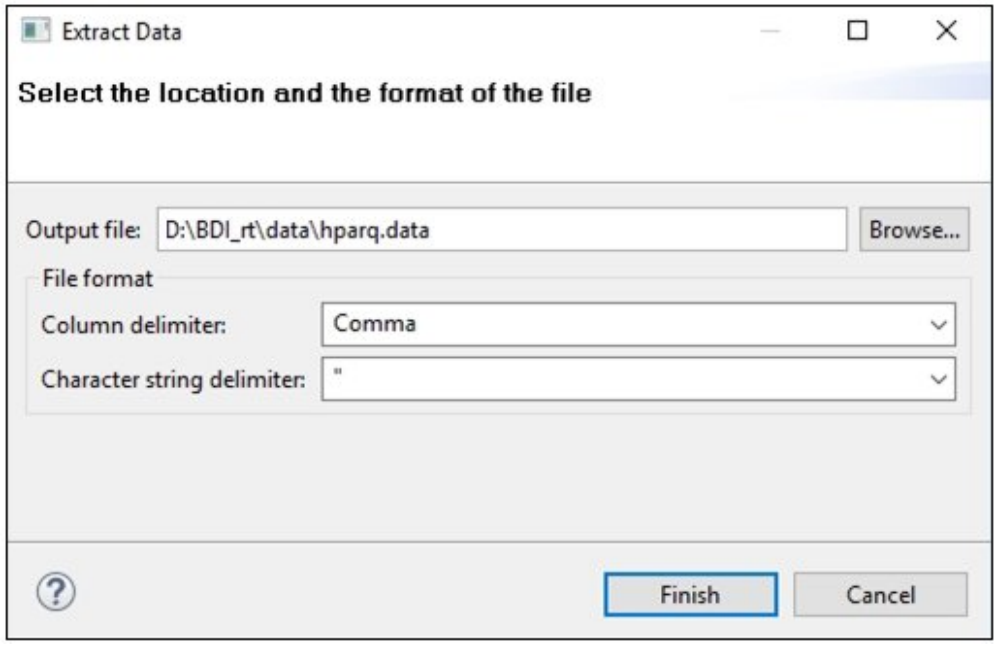
Sample Contents

The **Sample Contents** submenu option shows the first 50 records of the table in a scrollable result tab in the lower console pane of the iBDI window. View the column contents from this read only list.

Extract

The **Extract** submenu option should be used with care. This option extracts all of the records of the selected table from the Source system (Hive or RDBMS) into iBDI in record delimited format. There are no record limit numbers or limiting clauses. This is a strict data dump.

When the Extract submenu option is selected, the Extract Data dialog opens, as shown in the following image.



Perform the following steps:

1. In the Output file field, type or browse to the output folder and file.
2. From the Column delimiter drop-down list, select Comma, Semicolon, Space, Tab, or the vertical bar character (|).
3. From the Character string delimiter drop-down list, select double quote ("), single quote ('), or None.
4. Click *Finish*.

   The extraction process begins to the output file that is specified.

   When the data extract finishes, a message is displayed in the console window indicating the number of records that were written. For example:

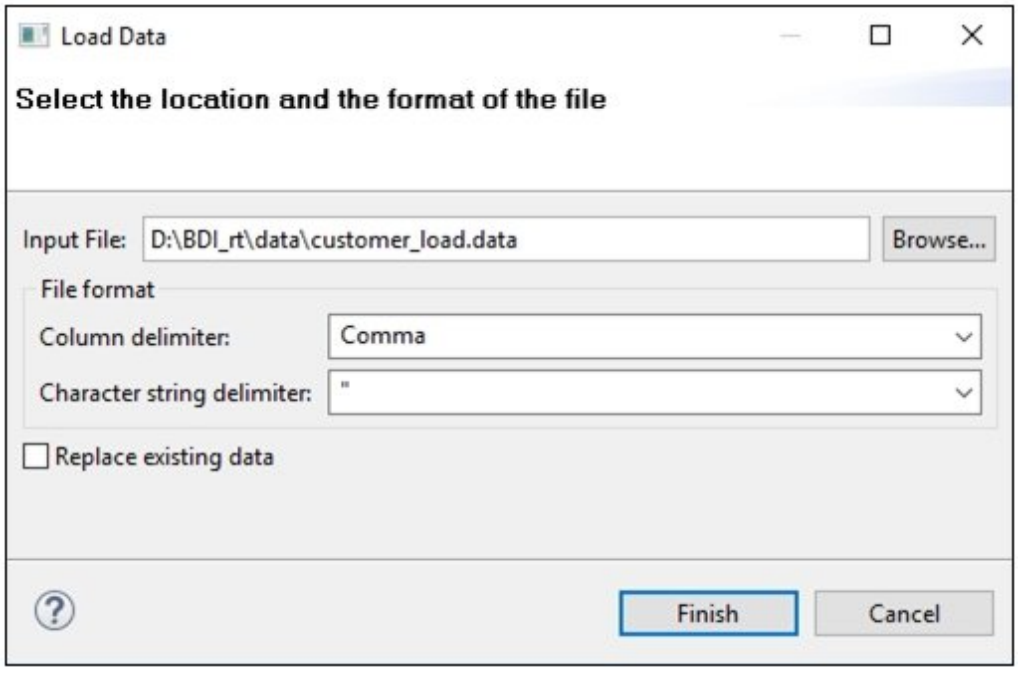
```
Extracting `customers`...
Data extraction was successful.
12435 row(s) extracted.
```

Load

The **Load** submenu option requires that a table already exist in the database with the specified name and that the columns in the table match the format of the columns in the incoming data.

Right-click the table in the Data Source Explorer for data loading, select *Data* from the context menu, and then click *Load*.

The Load Data dialog opens, as shown in the following image.



Perform the following steps:

1. In the Input file field, type or browse to the input folder and file.
2. From the Column delimiter drop-down list, select Comma, Semicolon, Space, Tab, or the vertical bar character (|).
3. From the Character string delimiter drop-down list, select double quote ("), single quote ('), or None.
4. Select or deselect the *Replace existing data* check box according to your requirements.
5. Click *Finish*.

   The data load process begins from the file location to the selected database schema and table name.

When the data finishes uploading from iBDI to the destination database, a message is displayed in the console window. For example:

```
Loading "customerxxx"...
Data loading was successful.
12435 row(s) loaded.
```

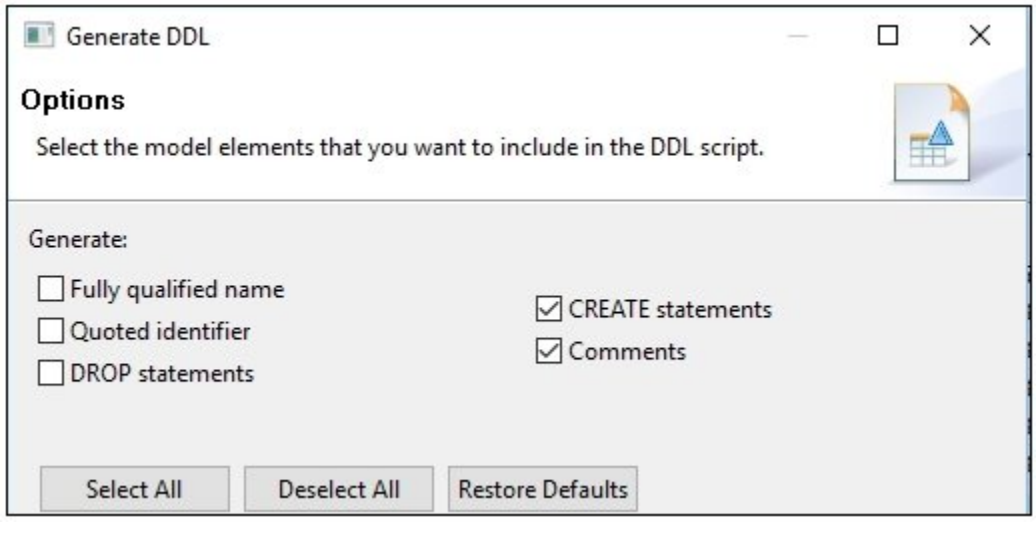
Replace

If the **Replace** submenu option is selected, then a delete will be issued against the table for the current contents. If Replace is not selected, then the data will attempt to be loaded as is. If there are duplicates in the data, then the underlying database validation will generate an exception and stop the job if constraints exists against duplicates. iBDI will display a message and the underlying database exception, as shown in the following example:

```
Loading "customerxxx"...
Data loading was successful.
0 row(s) loaded.
12435 row(s) could not be loaded.
com.mysql.jdbc.exceptions.jdbc4.MySQLIntegrityConstraintViolationException:
Duplicate entry '12435' for key 'PRIMARY'
```

Using the Generate DDL Wizard

The Generate DDL (data definitions) wizard allows you to export a model from any RDBMS and iBDI will translate it to HQL and create the tables in Hive. When *Generate DDL* is selected from the context menu in the Data Source Explorer, the Generate DDL wizard (Options pane) opens, as shown in the following image.

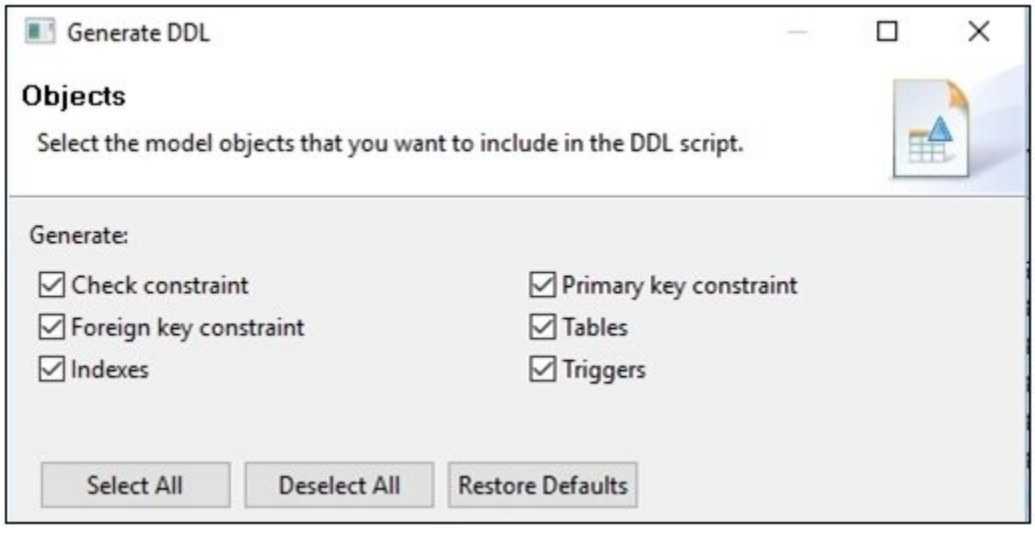


The following table lists and describes the supported model elements that you can select to include in the DDL script.

Model Element	Description
Fully qualified name	Use fully qualified names, including database, owner, and object.
Quoted identifier	Enable the quoted_identifier option that allows the interpretation of delimited strings to be changed.
DROP statements	Generate SQL DROP statements, which remove an existing object from the database.
CREATE statements	Generate SQL CREATE statements. Select the options associated with the DDL to CREATE or DROP, fully qualified names, quoted identifiers (special characters or character combinations matching SQL key words or non standard column names).
Comments	Generate comments.

After selecting the model elements, click *Next* at the bottom of the Generate DDL wizard to continue.

The Objects pane opens, as shown in the following image.

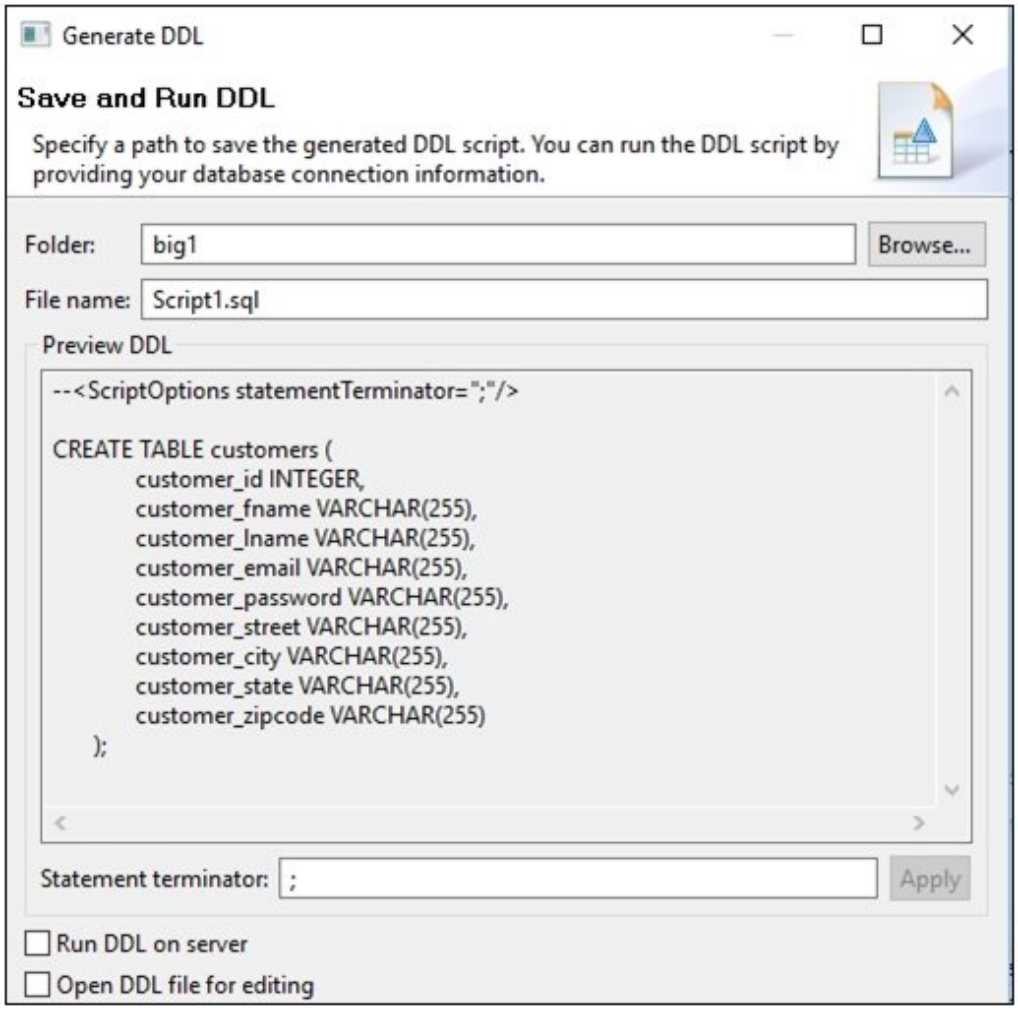


The following table lists and provides the scope of the supported model objects that you can select to include in the DDL script.

Model Object	Scope
Check constraint	Database, Table
Foreign key constraint	Database, Table
Indexes	Database, Table
Primary key constraint	Database, Table
Tables	Database, Table
Triggers	Database, Table
Views	Database **Note:** The *Views* model object is shown only when you are using Hive database connections.

Select the model objects to include in the DDL script, which can be all or only one of the objects in this pane. Click *Next* at the bottom of the Generate DDL wizard to continue.

The Save and Run DDL pane opens, as shown in the following image.



Perform the following steps:

1. In the Folder field, enter a project folder or click *Browse* to select the folder.
2. In the File name field, accept the default or enter a unique name.
3. In the Preview DDL area you can preview the generated DDL script.
4. In the Statement terminator field, accept the default terminator character or specify a different character to use.
5. Selecting the Run DDL on server checkbox allows you to run the DDL script against a connection profile.

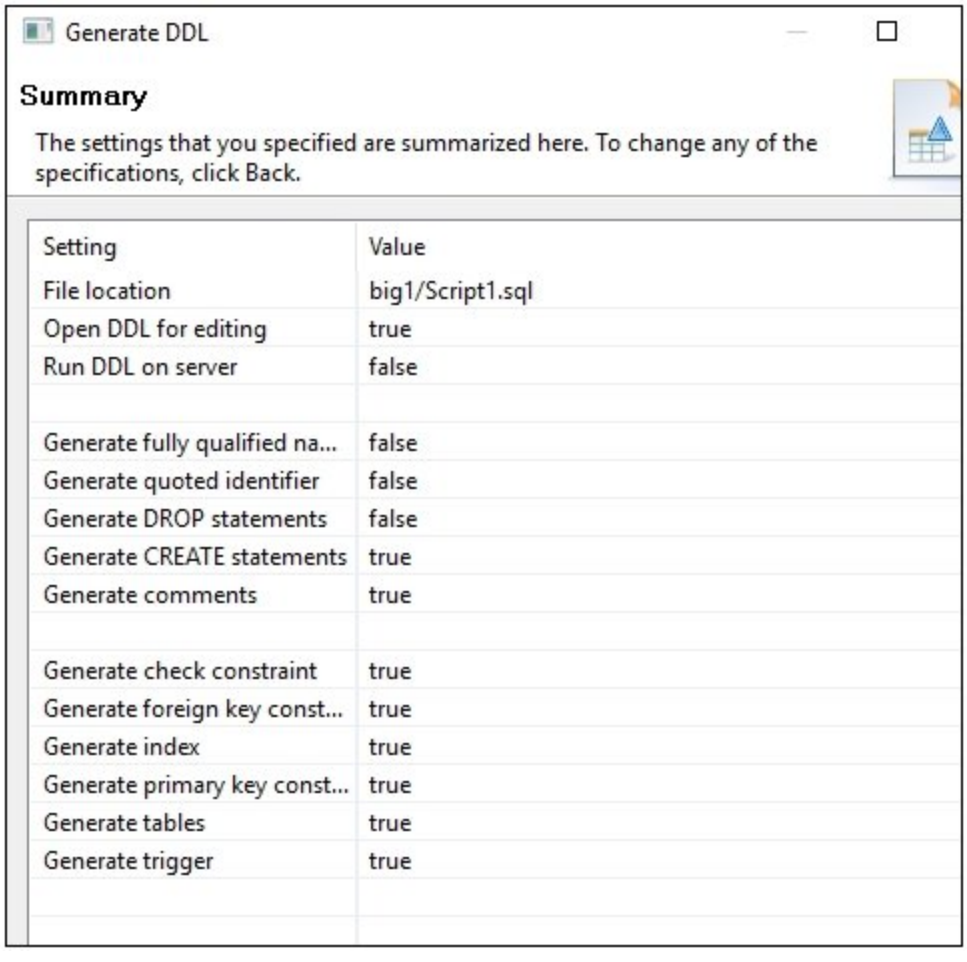
6. Selecting the Open DDL file for editing checkbox allows you to open the DDL script in SQL Scrapbook.

The current project name is automatically entered in the Folder field. You can select a different project by clicking *Browse*.

Running the DDL script will create the table in the current schema. Since the DDL is based on a current table, this can cause loss of data. Use the *Open DDL file for editing* checkbox to modify the DDL script by changing the name or other components (add or remove columns, change the field name or type).

Select the *Open DDL file for editing* checkbox and then click *Next* at the bottom of the Generate DDL wizard.

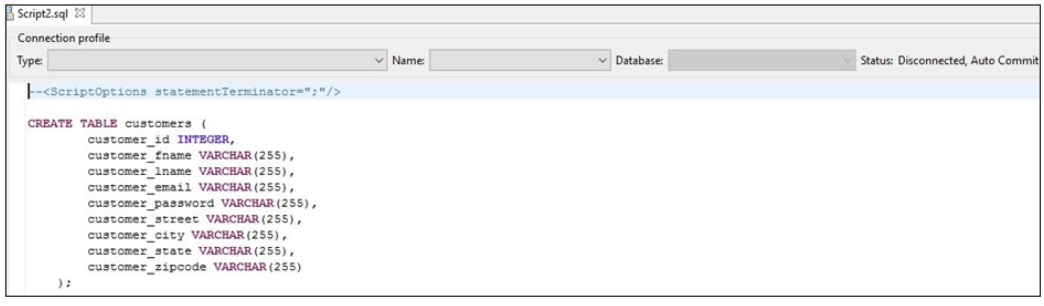
The Summary pane opens, which provides a summary of all the settings you configured in the Generate DDL wizard.



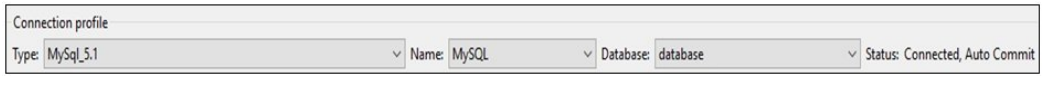
Setting	Value
File location	big1/Script1.sql
Open DDL for editing	true
Run DDL on server	false
Generate fully qualified na...	false
Generate quoted identifier	false
Generate DROP statements	false
Generate CREATE statements	true
Generate comments	true
Generate check constraint	true
Generate foreign key const...	true
Generate index	true
Generate primary key const...	true
Generate tables	true
Generate trigger	true

Click *Finish* at the bottom of the Generate DDL wizard.

The DDL script opens as a new tab in the iBDI perspective, as shown in the following image.



The data source Type and Name fields are initially blank. Select a connection profile that exists in the Data Source Explorer from the Type drop-down list.



If a connection profile does not have defined connection, then the Name and Database fields are blank. Select a valid type to continue.

The DDL script is saved in the DDL subfolder of the current iBDI project. Right-click the DDL script and select *Execute SQL Files* from the context menu to execute the script.

Note: The Generate DDL wizard creates standard ANSI SQL. Many database systems will not accept this dialect. As a result, the SQL must be edited before saving or executing to correctly format the SQL to the database standard.

Chapter 6

Understanding Pipelines

This chapter introduces pipelines and describes how to configure pipelines using iWay Big Data Integrator (iBDI).

In this chapter:

- Understanding Apache Spark
- Understanding a Pipeline
- Using the Pipeline Builder
- Building Your First Pipeline
- Configuring Apache NiFi

Understanding Apache Spark

Apache Spark is a distributed computing framework implemented along with Apache Hadoop to provide a wider range of functionality than is available in traditional MapReduce. The organizing concept is the DataFrame, a Dataset (distributed collection of data) organized into named columns. It is similar in concept to a relational database table, but has built in operations available.

Understanding a Pipeline

A *pipeline* can be thought of as a chain of stages that are connected from a *source* to a *target*. Data is transferred between stages in the pipeline. Each stage may perform a specific operation on the data (for example, *Show* or *Transform*).

A pipeline may have multiple *compute* stages between a source and a target. Each compute stage and the target have an input and an output schema associated with them. The source has no input schema, as the input schema is available from the data source itself. Because each node has an associated schema, intelligent operations can be performed on the data for the stage, such as validation, building expressions, and Where clauses.

Pipelines can be used for any number of operations, including data transformation, in memory computing, data cleansing with iWay Data Quality Server (DQS), modeling with RStat, and other operations.

A pipeline has a single execution path, without branching or looping. Pipelines can be chained together, so that the target stage for one pipeline becomes the source stage for the next pipeline for multiple or complex operations on data.

Source Stages

A *source* stage brings data from outside the pipeline into the pipeline for processing in subsequent stages. If the source is already in Hive, then transformations can be performed on the source object. Other sources can use a separate compute stage for actions or operations on part or all of the source data. A source can also consume the results of another pipelines target stage.

Source (Inbound data) use case selector:

- **Streaming**
  - **Stream.** Reads data from a stream into a DataFrame.
  - **Kafka.** Reads data from a Kafka queue into a DataFrame.
  - **Kafka Direct.** This new receiver-less direct approach has been introduced in Spark Version 1.3 to ensure stronger end-to-end guarantees.
  - **Nifi.** Reads data from an Apache NiFi stream from an output port into a DataFrame.
- **Defined**
  - **Hive.** Reads data from a Hive table into a DataFrame.
  - **HQL.** Executes HQL and loads it into a DataFrame.
  - **HDFS.** Reads data from a HDFS into a DataFrame.
- **Structured**
  - **RDBMS.** Reads data from an external RDBMS table into a DataFrame.

Compute Stages

A *compute* stage performs an action or operation on part or all of the source data. Multiple compute stages can be chained together to build a pipeline target result. Data can also changed using the Transformer tool.

The following table lists and describes the available compute stages in iBDI.

Compute Stage	Description
Cleanse	Submits a DataFrame to be cleansed in iWay Data Quality Server (DQS).
Coalese	Returns a new DataFrame with the exact number of partitions provided. This operation results in a narrow dependency. For example, if you go from 1000 partitions to 100 partitions, then there will not be a shuffle, Instead each of the 100 new partitions will claim 10 of the current.
Distinct	Returns a new DataFrame that contains only the unique rows from this DataFrame.
Drop	Returns a new DataFrame with a column dropped. This is a no-op if the schema does not contain a column name.
Filter	Filters rows using the provided SQL expression. For example: `age > 15`
Hive	Transforms a DataFrame by joining it with existing Hive tables.
HQL	Executes a HQL statement.
RStat Model	Scores the incoming data and adds a column to the DataFrame containing the prediction.
Order by	Returns a new DataFrame sorted by the given expressions. This is an alias of the Sort function.
Sample	Returns a new DataFrame by sampling a fraction of rows.
Save	Saves the DataFrame into Hive without stopping the pipeline.
Select	Selects an ordered list of columns from the DataFrame.
Show	Shows the first 20 records in a DataFrame.

Target Stages

A *target* stage is the final stage of a pipeline, which serves as a destination for the data from the pipeline. Pipeline targets can also function as sources for subsequent pipelines, depending on the type.

The following table lists and describes the available target stages in iBDI.

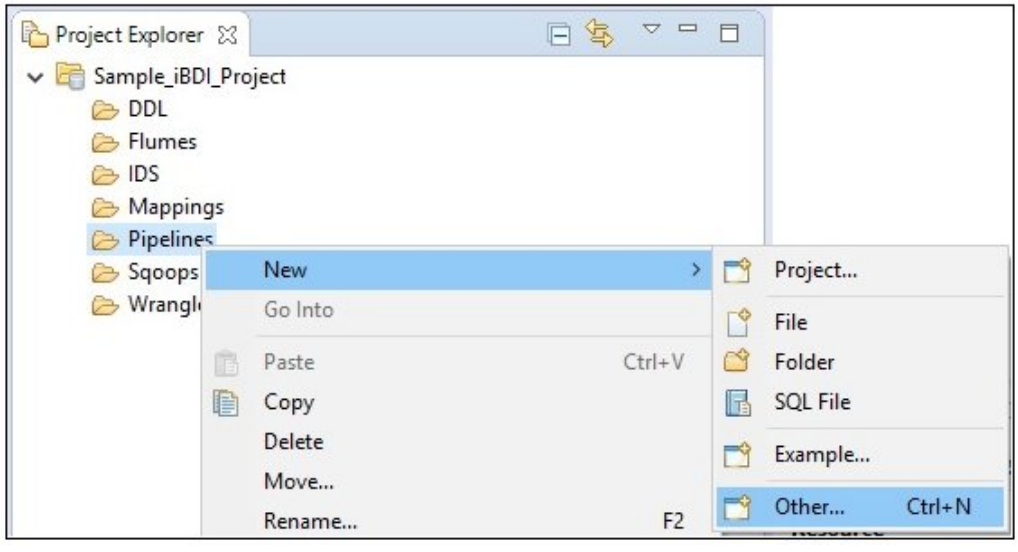
Target Stage	Description
Console	Displays the result of the pipeline operations in the Console pane of iBDI.
Stream	Kafka writes each row of the DataFrame into a Kafka topic.
Defined	❑ HDFS writes the DataFrame into HDFS. ❑ Hive writes the data to a Hive table. ❑ Hive CSV writes the DataFrame into HDFS as CSV.
Structured	RDBMS writes the data to an external table.

Using the Pipeline Builder

This section describes how to configure pipelines using the Pipeline Builder in iWay Big Data Integrator (iBDI).

Opening the Pipeline Builder

To open the Pipeline Builder, expand an available iBDI project node in the Project Explorer tab, right-click the *Pipelines* folder, select *New*, and then click *Other* from the context menu, as shown in the following image.



The New dialog opens.

Type *pipeline* in the field to filter the selection, select *Pipeline*, and then click *Next*.

You can also expand the *iWay Big Data Integrator* node in the New dialog and select *Pipeline*.

The Pipeline Builder can also be opened directly from the toolbar by clicking the *Pipeline Builder* icon. If you double-click an existing pipeline in the Pipelines folder (for example, pipe1.pipeline), the Pipeline Builder opens with the selected pipeline for editing.

The Pipeline Builder has two panes, as shown in the following image.

The left pane manipulates pipeline stages and the right pane displays parameters for the current stage, if that stage contains parameters.

Adding, Deleting, and Modifying a Stage

A new stage can be added to a pipeline by right-clicking on a stage and selecting *New Sibling* from the context menu or clicking the green plus (+) icon, as shown in the following image.

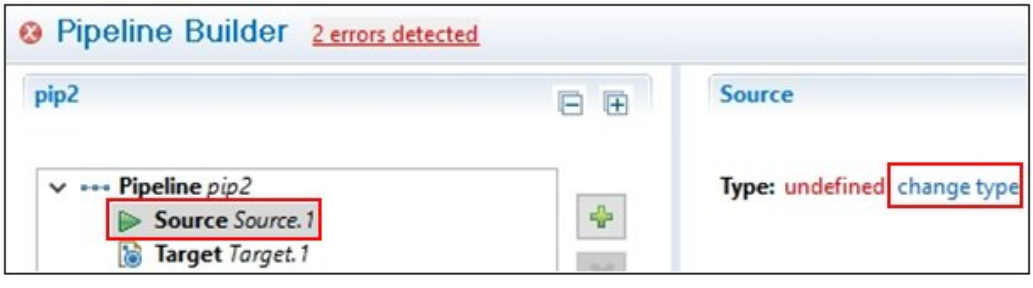
To delete a stage, click the red x icon, as shown in the following image.

The *Source* and *Target* stages cannot be deleted, they are required elements of a pipeline.

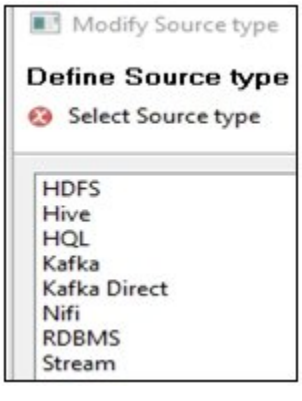
Pipeline stages can be added, deleted, or modified using the Component pane. Click on a Compute node to move it up or down in the sequence.

Adding, Deleting, and Modifying a Source

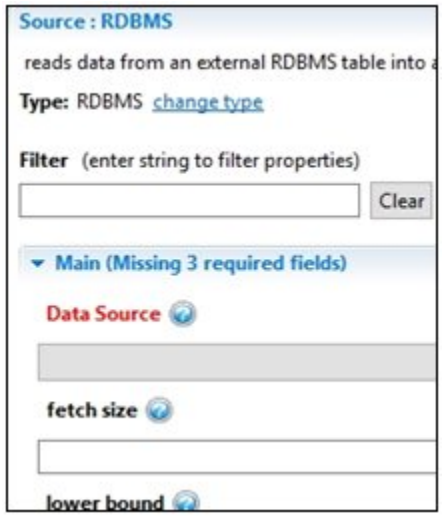
Add a source by clicking on a source entry in the left pane, then click *change type* in the right pane, as shown in the following image.



The Modify Source type dialog opens, as shown in the following image.



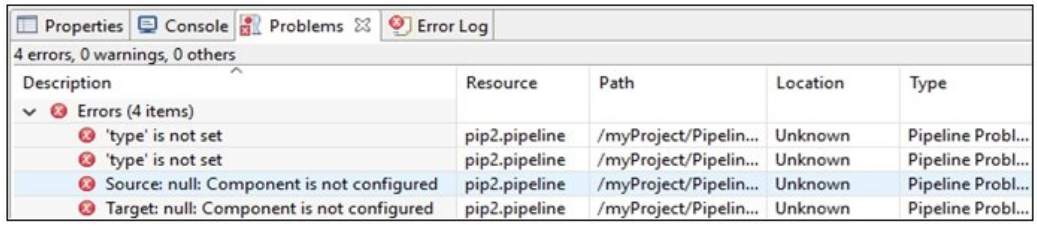
Select one of the available source types and then click *Finish* to continue. The right pane automatically opens the parameter view of the selected source type. For example, the following image shows RDBMS defined as a source type.



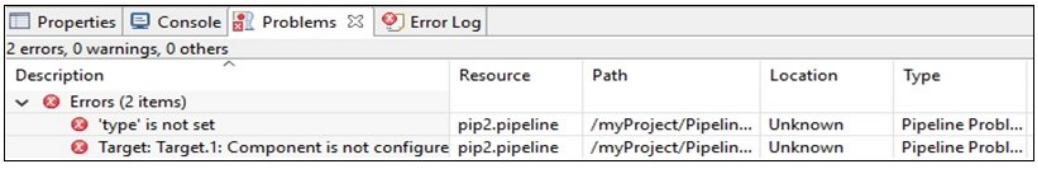
Using the Problems Pane

The Problems pane (or tab) in the lower part of the main window shows problems with the current pipeline. The status is changed when you click *Save* or *Save All*, as shown in the following image.

The following image shows the Problems pane before a pipeline source is configured:



The following image shows the Problems pane after a pipeline source is configured:



Click the *Save* icon, *File* and then *Save* from the main menu, or press *Ctrl*+*S*. The pipeline status is updated, so *Source* is not indicated as having a problem status.

Chaining Pipelines

Pipeline chains can be constructed by using the Target of one pipeline (for example, a Hive table) as the Source for another pipeline. One key advantage of pipeline chains is the ability to repeat processing stages in a controlled manner to a data set.

Troubleshooting Pipelines

Pipeline stages do not have dependencies on other pipeline stages. Many operations return a new DataFrame or a table. If one stage ends with an exception, then the entire pipeline stops processing.

In many cases, an error in a pipeline stage can be corrected and the pipeline can be run again from the start, but if an earlier pipeline stage persisted data to Hive or a similar operation, then the pipeline cannot be restarted from the beginning without duplication of data.

Recommended Technique

For deployed jobs, review the Console log for the pipeline in question. In the following example, the column used in an *OrderBy* stage does not exist in the Source:

```
11/23/2016 10:43:17.333 [INFO] Exception in thread "main"
org.apache.spark.sql.AnalysisException:
Cannot resolve column name "sample_column" among (department_id, department_name);
```

Corrective Action

Open and edit the pipeline in the Pipeline Builder. Open the Compute Stage in question and change the column name to one matching the columns in the table. Save the pipeline, then use the Run Configurations facility to run the pipeline again.

If the pipeline is run with a CRON job or script, then review the exception log on the edge node to find the error.

Jobs can fail as a result of user error, disk error, system error, network error, or other types of errors. If the data is important, ensure that you always have a recovery or backup plan in case of issues.

Building Your First Pipeline

This section provides information that will help you get started with configuring a pipeline using iWay Big Data Integrator (iBDI).

This use case demonstrates how structured (relational) data can be ingested through a pipeline. A data warehouse builder might use this example to understand how to model data within a Big Data context.

1. Select a *source* for the input.

   If you are familiar with Sqoop, then select *RDBMS* as the source. The right pane shows the properties of the selected object.

   Select a defined Data Source Profile from the drop-down list. Provide an integer column for *partition column - customer_id* and enter *sql* in the *sql - select * from customers* field.

2. Select a *compute* action.

   Examine the first 20 records and select *Show*.

3. Select a *target* schema type.

   Select *Console*.

4. Press *Ctrl*+S or click the *Save* icon on the toolbar.

5. Select *Run Configurations* from the main menu bar.

   The Run Configurations dialog opens.

6. Click the ellipsis button (…) next to the iWay Big Data Integrator Pipeline field.

   Right-click and select *New Configuration* from the context menu.

   The Create, Manage and Run Configurations dialog opens.

   Specify a name for the new pipeline configuration.

   In the Pipeline field, click *Browse* to open the current project. Expand *Pipelines*, open the *Pipelines* folder, then select the pipeline file to be deployed.

7. In the Deployment Build Target field, click *Browse*.

   Open the current project and then open the *Pipelines* folder.

8. For the Operation Type options, select *Publish* or *Deploy*.

When the job is to be run as a scheduled CRON job, select *Publish*. Otherwise, select *Deploy*.

9. Specify your connection information.

   Enter the host, port, user name, and password information where the iBDI edge node has been provisioned.

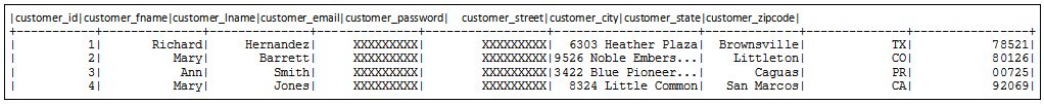
10. Specify your deployment information.

    For the path, enter the subfolder inside the iBDI provisioned edge node where this object will be located. If an existing name is used, then the object will be overwritten without warning in the client tool. The console will display the overwrite message.

11. Click *Apply* to save and apply your configuration settings.

12. Click *Run* to execute the pipeline configuration.

    The Console pane (deployments only) shows the job execution log. The output of the pipeline is shown inside the console. For example:



customer_id	customer_fname	customer_lname	customer_email	customer_password	customer_street	customer_city	customer_state	customer_zipcode
1	Richard	Hernandez	XXXXXXXXX	XXXXXXXXX	6303 Heather Plaza	Brownsville	TX	78521
2	Mary	Barrett	XXXXXXXXX	XXXXXXXXX	9526 Noble Embers...	Littleton	CO	80126
3	Ann	Smith	XXXXXXXXX	XXXXXXXXX	3422 Blue Pioneer...	Caguas	PR	00725
4	Mary	Jones	XXXXXXXXX	XXXXXXXXX	8324 Little Common	San Marcos	CA	92069

Configuring Apache NiFi

This section describes how to configure Apache NiFi. You can specify Apache NiFi as a streaming source stage in a Pipeline configuration.

For more information on installing Apache NiFi, see the *iWay Big Data Integrator Provisioning Guide*.

Procedure: How to Configure Apache NiFi

1. Enter the following URL in your browser to access your instance of Apache NiFi:

   `http://hostname:port/nifi`

   where:

   hostname

   Is where Apache NiFi is installed and running.

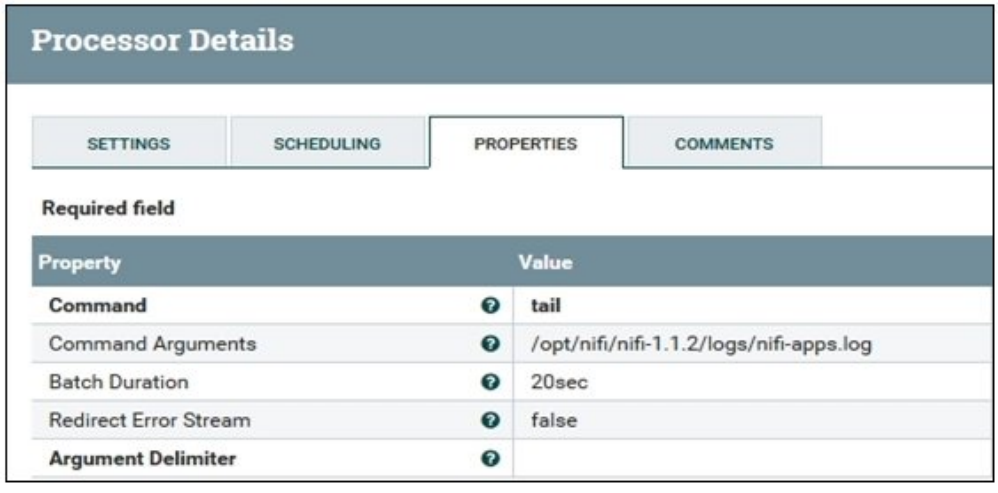
   port

   Is the port number that was assigned to be used by Apache NiFI during the installation process. For more information, see the *iWay Big Data Integrator Provisioning Guide*.

   The Apache NiFi web-based user interface (UI) opens.

2. In the Apache NiFi UI, drag a Processor object onto the canvas.
3. Select *ExecuteProcess* as the type for this Processor object.
4. Right-click the Processor object and select *Configure* from the context menu.

   The Processor Details dialog opens, as shown in the following image.

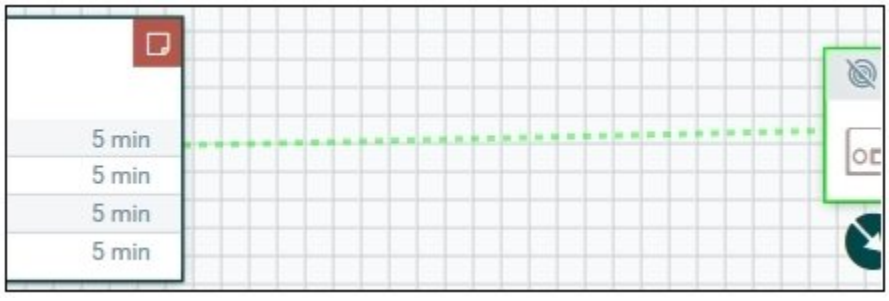


5. Specify the values for the configuration properties as listed in the following table.

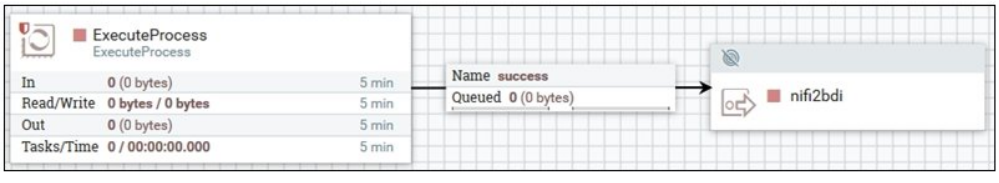
Property	Value
Command	tail
Command Arguments	/opt/nifi/nifi-1.1.2/logs/nifi-apps.log
Batch Duration	20sec
Redirect Error Stream	false

6. Drag an Output Port object onto the canvas, and name it *nifi2bdi*.

7. Click on the arrow icon in the Processor object and drag it to the Output Port object, as shown in the following image.



A connection object will be created to complete the FlowFile, as shown in the following image.



8. Click the red box icon within each object until it turns green, which indicates that the FlowFile has started.
9. Return to the iWay Big Data Integrator (iBDI) Eclipse GUI.
10. Create a new Pipeline where NiFI is specified as the Source.

    Specify the URL of your Apache NiFi instance (the server where it is running) and the name of the Output Port object that was created for Spark (*nifi2bdi*).
11. For testing purposes, select *console* as the Target of the Pipeline.
12. Save the Pipeline and create a new Run Configuration using the *iWay Big Data Integrator Publish* option.
13. Publish the Pipeline to the edge node.
14. Connect to the edge node using SSH or a similar tool.
15. Navigate to the folder that was created using the iBDI Eclipse GUI. For example:

    ```
    home/myuserid/bdipipelines/mynifi/
    ```

16. Change the directory to the /work directory:

    ```
    cd /work
    ```

17. Edit the *run.sh* script file using vi (visual editor) or a similar editor.
18. Add the following line under the *cp* statements section, which is located at the top of the script:

    ```
    cp ${HDM_HOME}/lib/nifi-*.jar .
    ```

    Note: Ensure that there is a space between `jar` and the period character (`.`)
19. Save the *run.sh* script file and run the script using the following command:

    ```
    ./run.sh
    ```

    The NiFi streaming should begin.

    Note: If an *Unsupported Major.Minor Version* exception should occur, then wrong Apache NiFi libraries have been deployed for the runtime Java version.

    Check the installed version of Java on the edge node by typing `which java` at the command line to show the folder where Java is installed.

    Check the version of Java by typing `java -version`.

Chapter 7

Understanding the Transformer

This chapter introduces the Transformer and describes how to configure transforms using iWay Big Data Integrator (iBDI).

In this chapter:

- iBDI Transformer Overview
- Getting Started With the iBDI Transformer

iBDI Transformer Overview

iWay Big Data Integrator (iBDI) provides the ability to transform data (Spark pipelines) through the Transformer tool. This transformed data can then be consumed by other enterprise applications.

Important Consideration:

In a pipeline, the Transformer is only supported for Hive table objects.

Getting Started With the iBDI Transformer

In the Pipeline Builder, select a new *Hive* source, and then click *Next*.

The Modify Source type dialog opens with the New Transformation pane, as shown in the following image.

The Connection Profile drop-down list provides all of the defined data sources (regardless of the project).

Select a Hive connection profile.

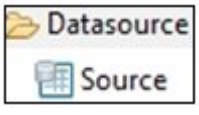
Click *Finish* at the bottom of the Modify Source type dialog to enable the Transformer tool.

In the right pane of the Pipeline Builder, the words *Manage Transform* appear, then the options for the transform. If any text does not fully display as a result of the screen size, you can resize the various panes accordingly by clicking and dragging the middle vertical bar.

Click *Edit Transform* to open the Transformer tool. Click *Reset Transform* to clear a previous transform and reset all values to the initial state.

The Transformer tool opens as a new tab in the main workspace area as a grid. A Palette is displayed in the right pane, which provides various objects that you can drag to the transform workspace area.

Click and drag *Source* from the Palette (under Datasource) to the workspace area.



The New Source Table dialog opens with the schemas for the database on the left. Open a schema by clicking the right arrow icon next to it or close a schema by clicking on the down arrow icon.

Open a schema and click on a table name, then in the right panel the fields of the table appear.

Fields from the selected table appear in the right pane.

Select a single field by clicking the check box next to the field name or click *Select All* to add all of the fields.

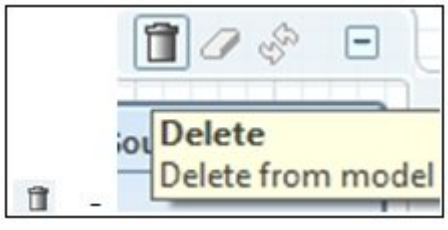
Note: Two extra fields appear in every table (*rec_ch* and *rec_modified_tsp*) These are used by iBDI for Change Data Capture (CDC), if enabled.

Select your fields as required, then click *Finish* to close the dialog and add the table to the workspace area.

The Source view appears in the workspace area, as shown in the following image.

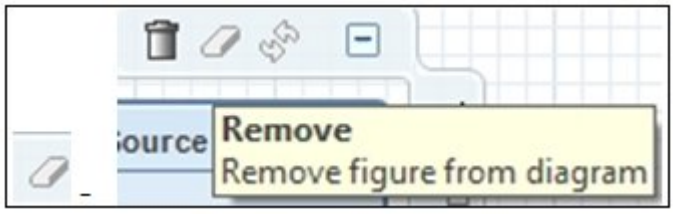
Move the cursor over the Source view to see the options that are available. The following options are standard for all views in the palette (grayed out options are unavailable):

- **Delete**

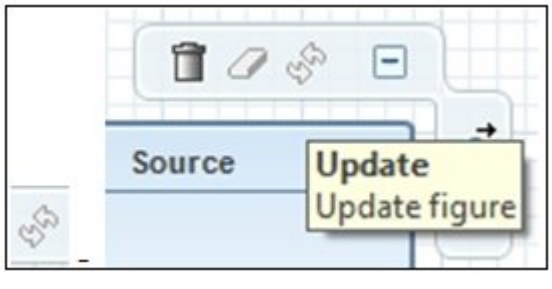


This option deletes this view from the model in the palette.

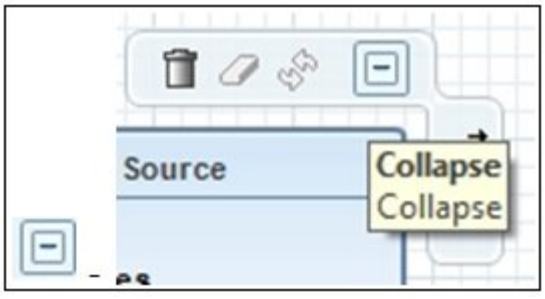
- **Remove**



- **Update**

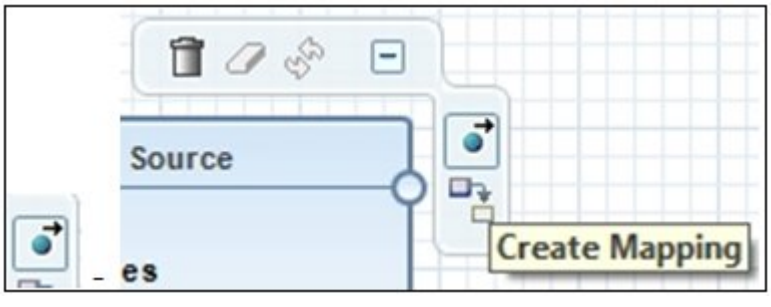


- **Collapse**

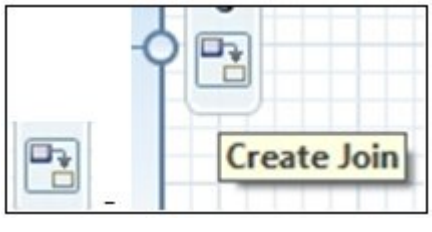


- **Expand**

- **Create Mapping**



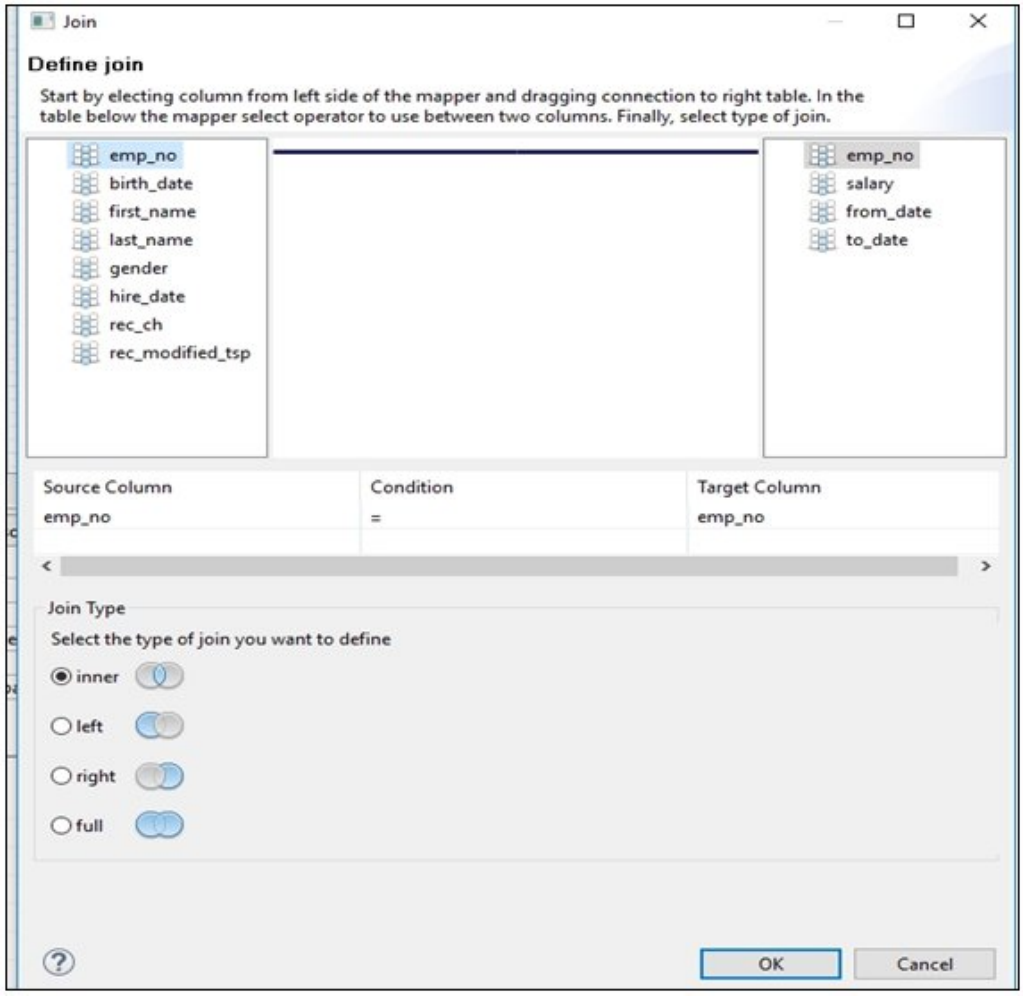
- **Create Join**



You can create connections, such as a Join or a Mapping, by clicking on the corresponding icon and moving the connection to the header of the next view object.

Click the *Create Join* icon and drag to the header of the next Source table. Do not drag on to the body, where the table name appears.

The Join dialog opens, as shown in the following image.



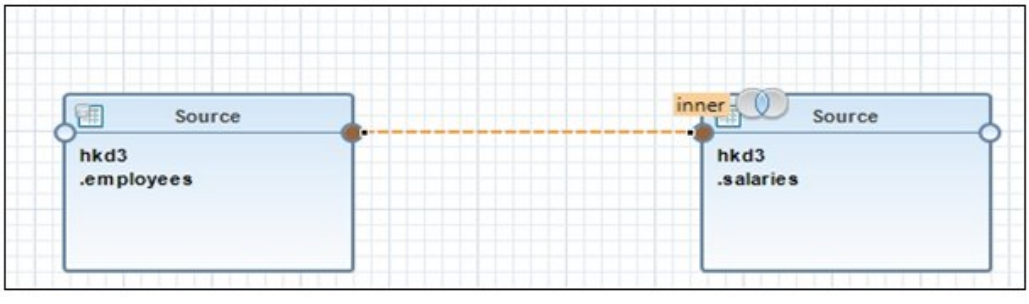
Click on a column name in the *Source Columns* area, which you want to join (map) to a target. Drag to the corresponding name in the *Target Columns* area. A mapping line is automatically drawn between the two columns. At the bottom of the source and target columns, the column names from source and target appear, with the *Condition* column available as a drop-down list. Select a condition from the drop-down list.

You can delete a Join by clicking on the mapping line and pressing *Delete*. You can also right-click the mapping line and select *Delete Mapping* from the context menu.

At the bottom of the Join dialog, select an available join type from the list (inner, left, right, or full). Be aware that some join types will consume more system resources (for example, processing time and disk space) than others. For more information on specific join types, see the following website:

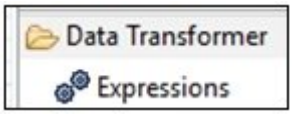
http://www.w3schools.com

Click *OK* to close the Join dialog. You can edit a Join by double-clicking the mapping line between the tables, which opens the Join dialog for editing.



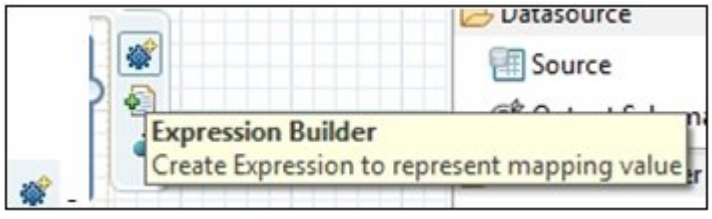
The Join appears on the model view with the Join type name and indicator appearing in the *Join to* table header.

To use the Expressions tool, click and drag *Expressions* from the Palette (under Data Transformer) to the workspace area.

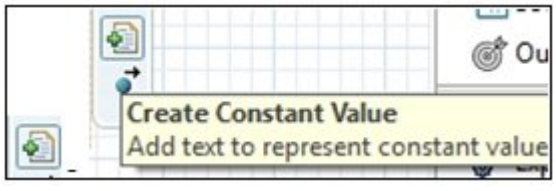


The Expressions tool enables you to add functional expressions into the target. The Expressions tool provides the following modes in addition to the standard model view tools:

- **Expression Builder**



- **Create Constant Value**



Mapping Fields to the Output Schema

The Output Schema is the endpoint of the transformation. It contains only those columns that have been explicitly added.

Perform the following steps to configure a simple mapping:

1. Click and drag the *Source* object from the Palette to the transform workspace area.
2. View the properties in the Properties dialog to obtain the data types of the columns in the source.
3. Position the Select tool on the object header (for example, Source or Expressions) and click the *Map* icon that appears above as the cursor hovers on the header.
4. Drag the *Map* icon to the Output Schema.

   A line will appear on the transform workspace area. When the Map icon is dropped in the Output Schema object, the Define Mappings dialog opens.

   The columns from the source Hive table appear in the left column and will be placed in the Target column to be reproduced in the target.
5. Click on an individual field and drag it to the target list or use the *Ctrl* key to select all fields and drag them to the target.

Complex Mappings

Multiple source objects can be added to the transform workspace area. At each step, map the columns from the source to appear in the output object. The Properties dialog for each object will show the type and length of each column. The Output Schema will contain the total columns for all source objects.

Joining Tables

Click the Join icon and drag the table to be joined to the target to create a new table merge. The Join dialog opens.

A Join should have fields of similar type and length to match the fields. Once the Join is created, the Join dialog will shown the columns mapped and allow the Join Condition (selection operator) to be changed by clicking on it and selecting another operator from the list.

Use the radio button to select the Join type to be performed. For more information, see the *SQL JOINs* topic on the following website:

http://www.w3schools.com

When a Join is defined, the type of Join appears in the header of the target table.

The Join tool exists only to create Joins between tables based on selected matching columns. Draw Join lines separately. All source objects added to the transform workspace area should be joined to each other and mapped to the Output Schema, since independent sources cannot co-exist with a Join.

Each Source table on the transform workspace area has to have the output fields mapped to the Output Schema in order to have columns appear in the Output Schema.

First, view the objects on the transform workspace area in the Properties dialog to understand the types and lengths. Then use the Join tool to join tables based on common affinities. Then use the Mapping tool for each table on the workspace area to draw lines between the individual source and the Output Schema. This allows for precise column selection.

General Expression Builder

Drag the Expression Builder icon to the transform workspace area and then join the Expression Builder to the first source table. This will link the metadata of the tables to the Expression Builder. Type the first letter of the schema name to obtain the field names as a select list in the following format:

```
schemaname.tablename.columname
```

The Help area at the bottom of the Expression Builder dialog contains entries for many expressions and operators. Simply type the expression or operator in the dialog.

Click the icon to enter a new constant value. Constants are mapped in the same way as fields to the Output Schema.

Where Clause Expression Builder

Click the Funnel icon on the Output Schema to open the Expression Builder in Filter mode. The word *Where* does not need to be typed. Select fields from one or more sources by using the following convention:

```
schemaname.tablename.columnname
```

The Help area at the bottom of the Expression Builder dialog contains entries for many expressions and operators. Simply type the expression or operator in the dialog.

HQL

Click the *HQL* tab at the bottom of the Transform tab to view the HQL that is generated by the Transformer as tables are joined and mapped.

Saving Your Work

When you have finished defining and configuring the transform, click the *Save* icon on the toolbar. The transform is saved in the pipeline and can be edited at any time. For frequently used projects, pipelines can also be copied between projects in the workspace.

Chapter 8

Creating a Sqoop Configuration

This chapter introduces Sqoops and describes how to create a Sqoop configuration using iWay Big Data Integrator (iBDI).

In this chapter:

- Creating and Executing a Sqoop Configuration
- Exporting Data Using Sqoop

Creating and Executing a Sqoop Configuration

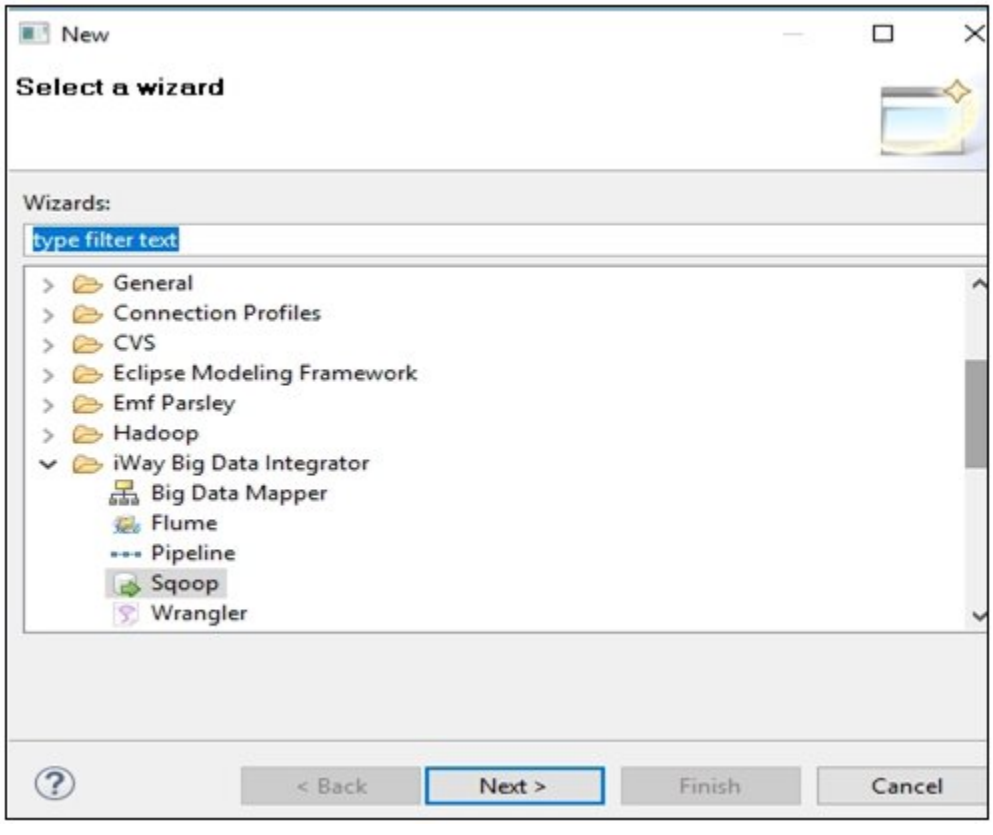
Apache Sqoop is designed for bulk data input from structured data sources, such as relational databases, into a HDFS. Sqoop reads records row by row in parallel and writes them in the selected file format into a HDFS. Sqoop usually splits by the primary key, or a defined primary key if no primary key is present.

Note: Sqoops are a non-Pipeline type of data configuration.

Procedure: How to Create a Sqoop Configuration

1. Expand an available iBDI project node in the Project Explorer tab, right-click the *Sqoops* folder, select *New*, and then click *Other* from the context menu.

The *New* dialog opens, as shown in the following image.

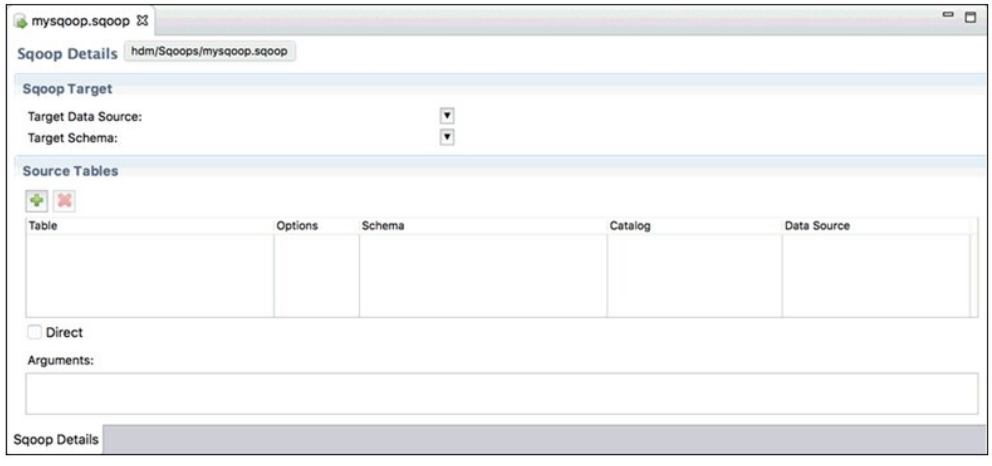


2. Type *sqoop* in the field to filter the selection, select *Sqoop*, and then click *Next*.

   The New Sqoop dialog opens.

3. Type a name for the new Sqoop in the Name field (for example, *mysqoop*) and click *Finish*.

The Sqoop opens as a new tab (mysqoop.sqoop) in the iBDI workspace, as shown in the following image.



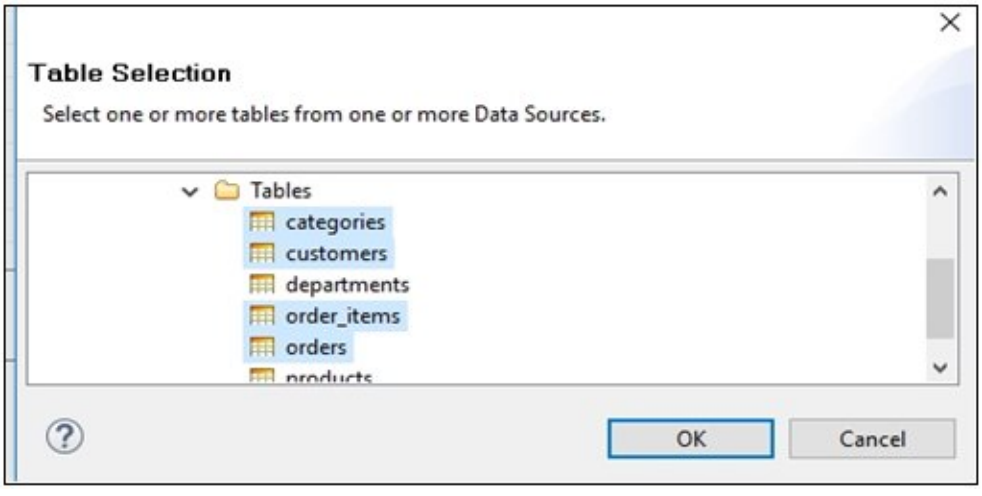
4. From the Target Data Source drop-down list, select an available Hive connection.

5. From the Target Schema drop-down list, select the schema into which your data will be imported.
6. Click *Add* (the green plus sign icon) to select the table(s) you want to import.

   To remove a table, select the table and then click *Delete* (the red X icon).

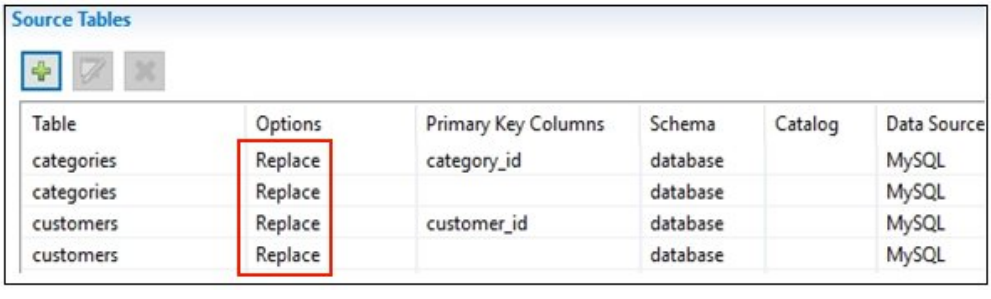
   Note: If no database connections have been defined, these fields will appear empty. Go to the Data Source Explorer pane and define the connections to the source and target. Close and then open the Sqoop tab to continue.

More than one table can be selected, as shown in the following image.



7. Select the required table(s) and click *OK*.

   The selected tables are now populated in the Source Tables area, as shown in the following image.

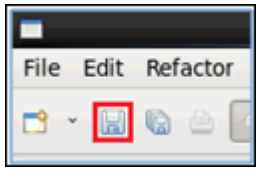


The following table lists and describes the available setting you can select for each table in the Options column.

Option	Description
Replace	Replaces the data if it already exists in the HDFS and Hive. Subsequent Sqoops of the same table will replace the data.

Option	Description
CDC	A key feature in iBDI, which executes Change Data Capture (CDC) logic. This means that subsequent executions of this Sqoop configuration will append only changes to the data. Two extra columns are added to the target table that enables CDC: ❑ **rec_ch.** Contains the hashed value of the record. ❑ **rec_modified_tsp.** Is the timestamp of when the record was appended.
Native	Allows advanced users to specify their own Sqoop options.

8. Click the *Save* icon or use the Ctrl+S shortcut to save your work.



Procedure: How to Execute a Sqoop Configuration

Once you have created a Sqoop configuration, a Sqoop file with a .sqoop extension is added to your *Sqoops* directory in your iWay Big Data Integrator (iBDI) project.

For more information, see *Defining Run Configurations* on page 147.

Exporting Data Using Sqoop

Sqoop can also be used to export data from Hive to relational tables. The table to contain the data must exist in the relational database, or the Sqoop will fail. You can use the Generate DDL wizard to create DDL statements to create the table (adjusting column lengths and types as required), and then execute the DDL statements from the Data Source Explorer. For more information, see *Using the Generate DDL Wizard* on page 77.

You can then create and execute the Sqoop export.

Run the DDL script to create the table on the database server. Check the Console to ensure that no errors have occurred and then look in the Data Source Explorer for the table. Select *Refresh* or press *F5* before checking the Data Source Explorer.

After the table is confirmed in the target database, create a new Sqoop configuration as described in *How to Create a Sqoop Configuration* on page 109. You can also click the following icon on the toolbar:

The Sqoop opens as a new tab in the iBDI workspace.

The Target Data Source will be the structured relational data source (such as MySQL, Postgres, and so on). Use the table schema defined during the configuration of the data source connection. The Source Tables will be Hive tables.

Click *Add* (the green plus sign icon) to select the table(s) you want to add. To remove a table, select the table and then click *Delete* (the red X icon).

Once a table is added to the list, click a table name, and then click the following Edit icon to only copy the selected columns:

This option allows the possibility of creating tables based on sub-selections of main tables. The number of columns in the Output target schema and the columns in the target database must match. If you are performing a sub-select, ensure that the columns have the correct alignment.

Click the *Save* icon or use the Ctrl+S shortcut to save your work. Use the Run Configurations facility to deploy or publish the Sqoop configuration to the edge node. For more information, see *Defining Run Configurations* on page 147.

Chapter 9

Creating a Flume Configuration

This chapter introduces Flumes and describes how to create a Flume configuration using iWay Big Data Integrator (iBDI).

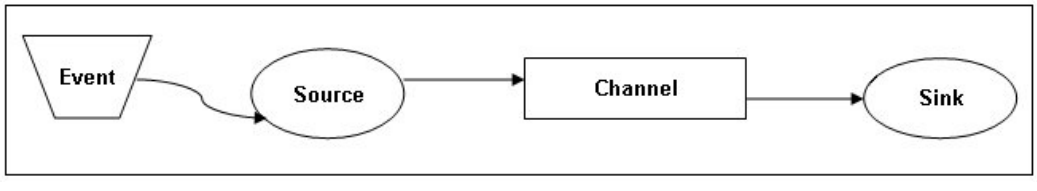
In this chapter:

- Understanding Flume
- Creating and Executing a Flume Configuration
- Flume Parameters Reference

Understanding Flume

Flume is a distributed service for collecting and moving large amounts of streaming event type data. It has fault tolerant options with recovery and failover mechanisms. Flume has three components, a *Source* that consumes event data, a *Channel* that stores the data until a *Sink* moves the data to a persisted location or another Flume source.

Flume can be combined with Kafka for *Flafka*, a sophisticated technique to capture, trace, and persist streaming event data.



Note: Flumes are a non-Pipeline type of data configuration.

Creating and Executing a Flume Configuration

This section describes how to create and execute a Flume configuration.

Procedure: How to Create a Flume Configuration

1. Expand an available iBDI project node in the Project Explorer tab, right-click the *Flumes* folder, select *New*, and then click *Other* from the context menu.

   The New dialog opens.

2. Type *flume* in the field to filter the selection, select *Flume*, and then click *Next*.

The New Flume dialog opens.

3. Type a name for the new Flume in the Name field and click *Next*.

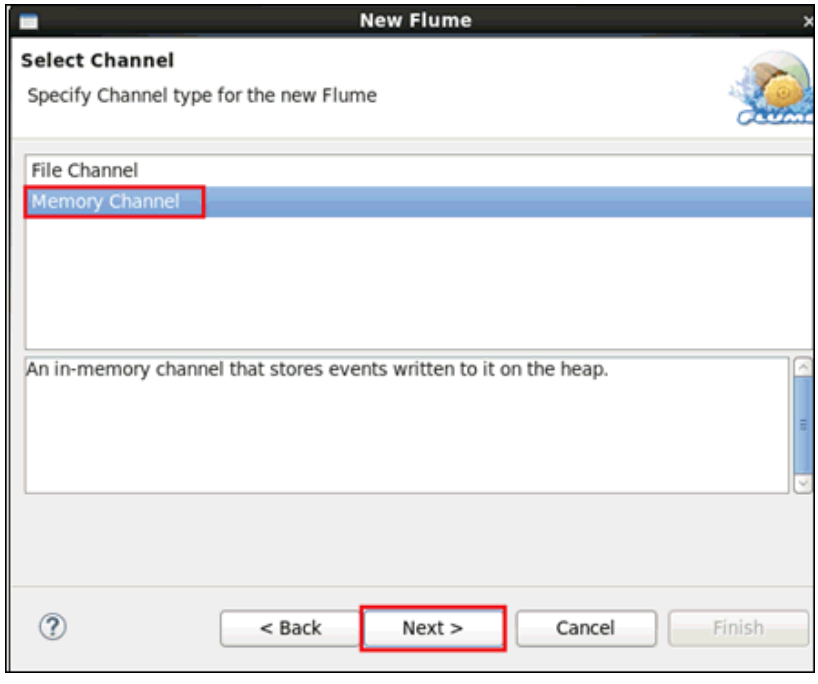
   The Select Source pane opens.

   The following table lists and describes the available Source types that you can select for the Flume configuration.

Source	Description
Avro	Listens on an Avro port and consumes Avro client streams.
HTTP	Accepts Flume events through HTTP POST. The default implementation has a JSON handler.
Exec	Runs a Linux command on startup and expects that process to continually produce data on standard output. If the process exits for any reason, then the source exits and will produce no further data.
SpoolDirSource	Listens for files in a spooling directory and parses events out of the files. The files are renamed or deleted after input into the source. All file names must be unique for this Source type.
JMS	Reads messages from a JMS destination, such as a queue or topic. Flume vendor JMS dependencies must be resolved before using this Source type. Data may require a *converter* from JMS format to Flume event format.

4. Select a Source type and click *Next*.

The Select Channel pane opens, as shown in the following image.



Only the File Channel has recovery and data loss recovery possibilities. Durable channels persist the data so it can be restored if a machine or disk failure occurs. Memory channels can be faster.

- **File Channel.** Used when the data must be protected against loss. File channels lock the directories, so each Flume channel should have an explicit path, preferably on different disk drives.
- **Memory Channel.** Designed for high throughput and no backup in case of failures.

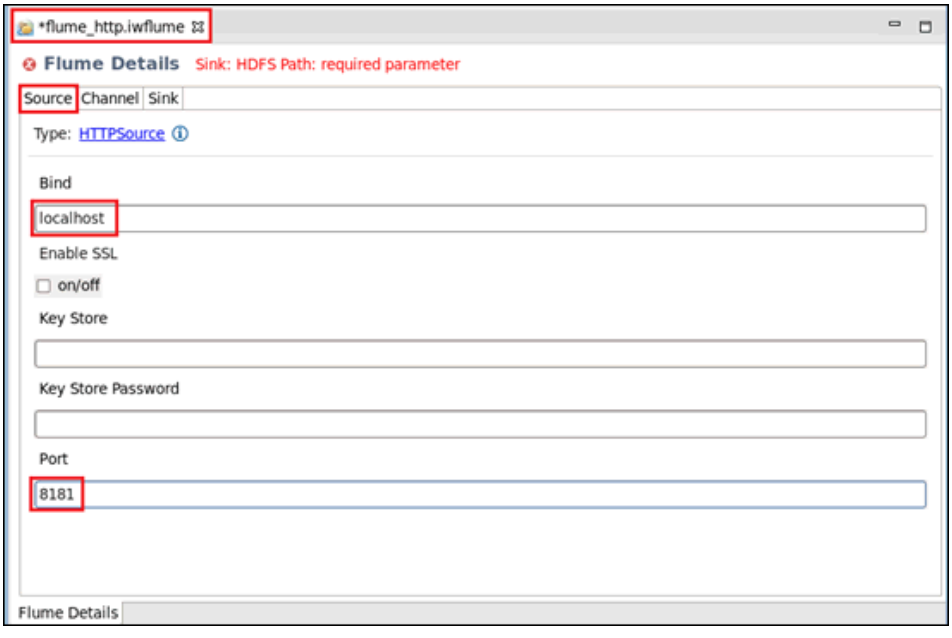
5. Select a channel type and click *Next*.

   The Select Sink pane opens.

   The HDFS sink writes data to HDFS as text or sequence file. The files can be rolled (close current and open new) at periodic intervals. Compression is also supported. The HDFS path name can contain escape sequences that will be replaced by the sink to generate the directory and file name to store events.

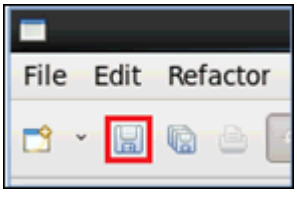
6. Select *HDFS*, *Avro*, or *Kafka*, and then click *Finish*.

The Flume opens as a new tab (for example, flume_http.iwflume) in the iBDI workspace where specific Flume details can be configured, as shown in the following image.



Three tabs are provided in the Flume Details pane (Source, Channel, and Sink), which can be configured. The Source tab is selected by default.

7. Click the *Save* icon or use the Ctrl+S shortcut to save your work.



Procedure: How to Execute a Flume Configuration

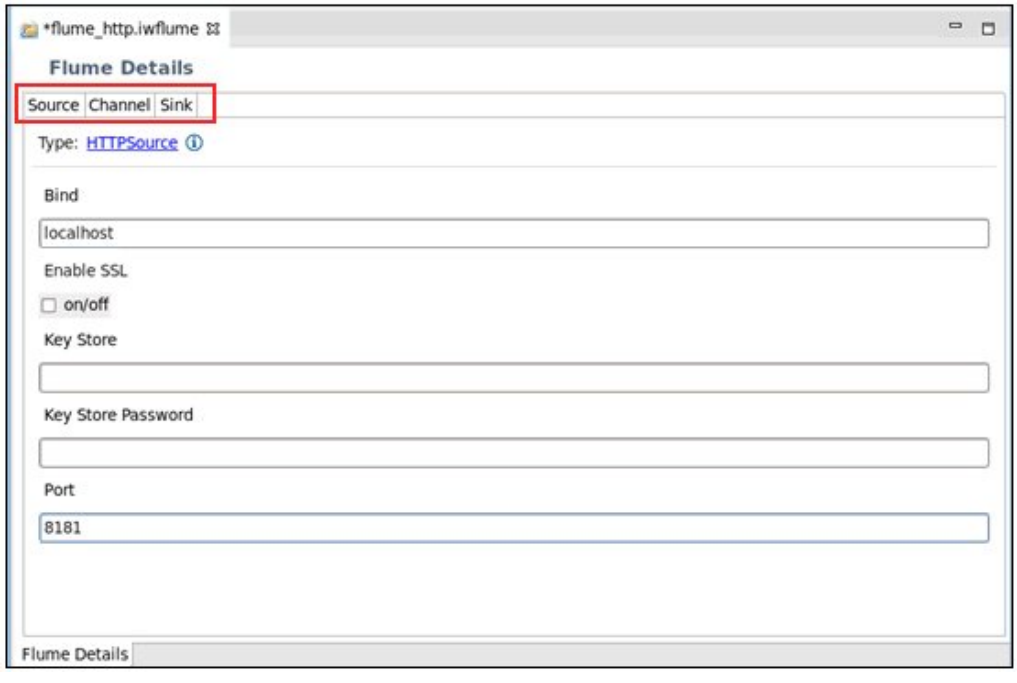
Once you have created a Flume configuration, a Flume file with a .iwflume extension is added to your *Flume*s directory in your iWay Big Data Integrator (iBDI) project.

Note: When executing a Flume configuration, ensure that you are selecting and right-clicking the *iWay Big Data Integrator Publish* option and selecting *New* from the context menu in the Run Configurations facility, and not the *iWay Big Data Integrator Deploy* option.

For more information, see *Defining Run Configurations* on page 147.

Flume Parameters Reference

This section provides a reference for the Flume parameters that are available in the three tabs of the Flume Details pane (Source, Channel, and Sink), as shown in the following image.



Common Parameters

The following parameters are common to HTTP and Avro source types:

- Enable SSL check box (on | off)
- Key Store
- Key Store Password
- Key Store Type
- Port

Source Parameters

This section provides a reference for the Flume parameters that are available in the Source tab of the Flume Details pane.

Avro

- Bind
- Compression Type (blank or deflate)
- Threads
- Integer max number of worker threads to accept from clients

HTTP

- Bind

Exec

- Batch Size
- Batch Timeout
- Charset
- Command
- Log Standard Error check box (on | off)
- Restart check box (on | off)
- Restart Throttle
- Shell

SpoolDirSource

- Batch Size
- Delete Policy
- Serializer
- File Header Key

- File Suffix
- Ignore Pattern
- Input Charset r
- Is File Header check box (on | off)
- Tracker Directory

ExecSource

- Batch Size
- Batch Timeout
- Charset
- Command
- Log Standard Error check box (on | off)
- Restart check box (on | off)
- Restart Throttle
- Shell

JMSSource

- Batch Size
- Connection Factory
- Converter Charset
- Converter Type
- Destination Name r
- Destination Type r
- Initial Context Factory r
- Message Selector
- Password File

- Provider Url r
- User Name

Channel Parameters

This section provides a reference for the Flume parameters that are available in the Channel tab of the Flume Details pane.

Memory Channel

- Byte Capacity
- Capacity
- Keep Alive
- Transaction Capacity

File Channel

- Backup Checkpoint Dir
- Checkpoint Interval
- Data Dirs
- Keep Alive
- Max File Size
- Minimum Required Space
- Transaction Capacity
- Use Dual Checkpoints check box (on | off)
- capacity
- CheckpointDir

Sink Parameters

This section provides a reference for the Flume parameters that are available in the Sink tab of the Flume Details pane.

Avro Sink

- Batch Size
- Compression Level
- Compression Type
- Connection Timeout
- Exclude Protocols
- HDFS Path check box (on | off)
- Host Name r
- Max IO Workers
- Port r
- Request Timeout
- Reset Connection Interval
- SSL check box (on | off)
- Truststore
- Truststore Password
- Truststore Type

Kafka Sink

- Broker List
- Topic

HDFS Sink

- Batch Size
- Call Timeout
- CodeC
- File Prefix

- File Suffix
- File Type
- Idle Timeout
- In use Prefix
- In use Suffix
- Kerberos key tab
- Kerberos Principal
- Max Open files
- Path
- Proxy User
- Roll Count
- Roll Interval
- Roll Timer Pool Size
- Round check box (on | off)
- Round Unit
- Round Value
- Threads Pool Size
- Time Zone
- Use Local TimeStamp check box (on | off)
- Serializer

Flume File Name Escape Sequences

The following table lists and describes the Flume file name escape sequences.

Escape Sequence	Description
%{host}	Substitute value of event header named *host*. Arbitrary header names are supported.
%t	Unix time in milliseconds
%a	Locale's short weekday name (Mon, Tue, etc.)
%A	Locale's full weekday name (Monday, Tuesday, etc.)
%b	Locale's short month name (Jan, Feb, etc.)
%B	Locale's long month name (January, February, etc.)
%c	Locale's date and time (Thu Mar 3 23:05:25 2005)
%d	Day of month (01)
%e	Day of month without padding (1)
%D	Date, same as %m/%d/%y
%H	Hour (00...23)
%I	Hour (01...12)
%j	Day of year (001...366)
%k	Hour (0...23)
%m	Month (01...12)
%n	Month without padding (1...12)
%M	Minute (00...59)
%p	Locale's equivalent of AM or PM
%s	Seconds since 1970-01-01 00:00:00 UTC
%S	Second (00...59)

Escape Sequence	Description
%y	Last two digits of year (00...99)
%Y	Year (2010)
%z	+hhmm numeric timezone (for example, -0400)

Chapter 10

Creating a Mapping Configuration

This chapter introduces mappings and describes how to create a mapping configuration using the Big Data Mapper in iWay Big Data Integrator (iBDI).

In this chapter:

- Big Data Mapper Overview
- Creating and Configuring a New Mapping

Big Data Mapper Overview

It is recommended to use Spark pipelines to transform data because of the variety of source inputs, outputs, and the operations that are permitted. A mapping configuration created using the Big Data Mapper is a one stage procedure. A configured pipeline enables multiple processing stages. The Big Data Mapper facility in iBDI is used to transform data contained in Hive tables.

Understanding the Big Data Mapper

The Big Data Mapper is a column-oriented data mapping engine. The Transformer in the Pipeline is a table-oriented mapping engine that uses newer features of Spark to perform transformations.

In the Big Data Mapper, a source table is dropped onto the workspace and all of the columns for that table appear with the source table. Drag a Target object onto the workspace, and then map the individual fields from the source to the target.

You can join two sources by first dragging the Source objects onto the workspace, then using the Join icon from the tool menu or the Palette to drag and join similar fields.

The Big Data Mapper offers the following target table options:

- Existing table
- Input Document Specification (IDS) file
- New table

Note: Mappings are a non-Pipeline type of data configuration. The Big Data Mapper only transforms Hive tables.

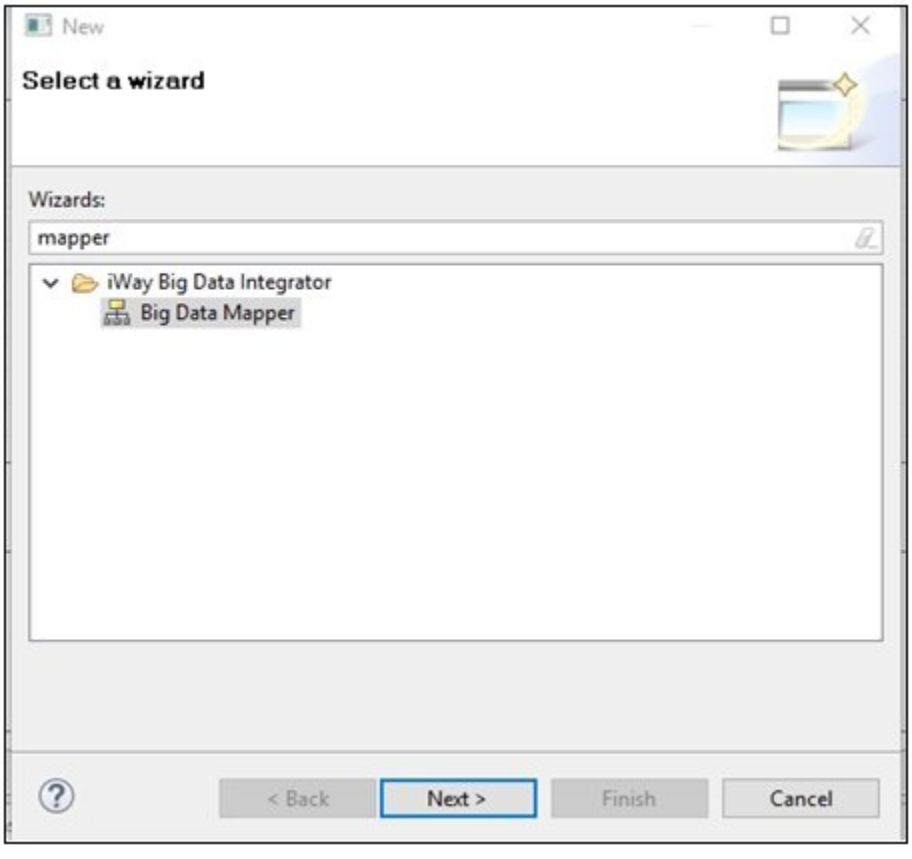
Creating and Configuring a New Mapping

This section describes how to create and configure a new mapping.

Procedure: How to Create and Configure a New Mapping

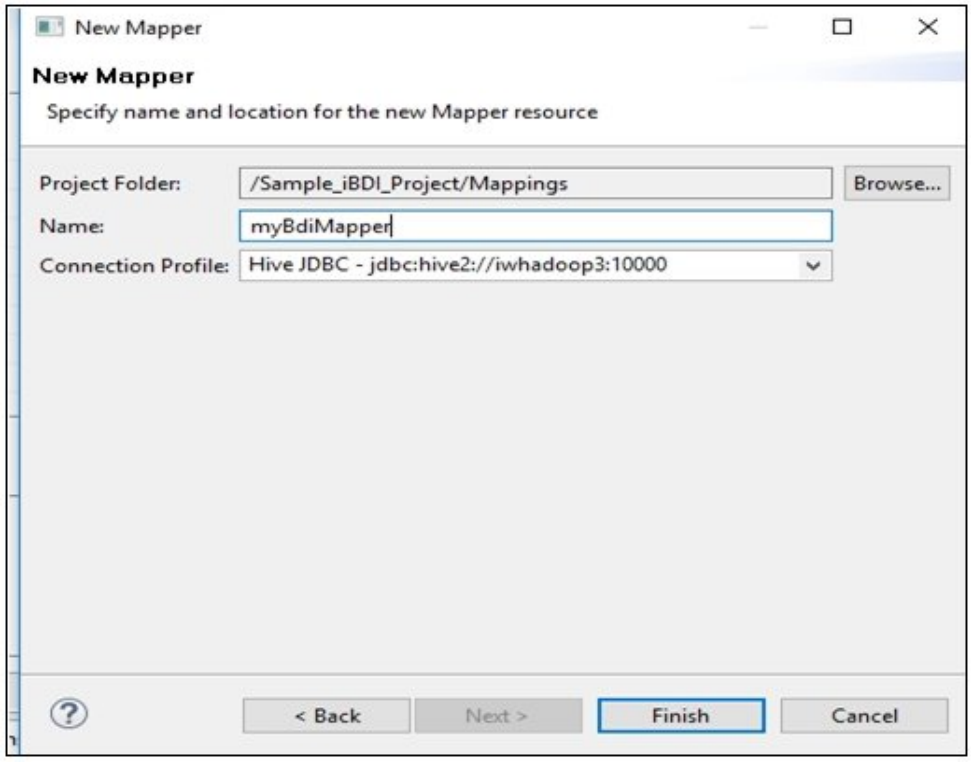
1. Expand an available iBDI project node in the Project Explorer tab, right-click the *Mappings* folder, select *New*, and then click *Other* from the context menu.

   The New dialog opens, as shown in the following image.



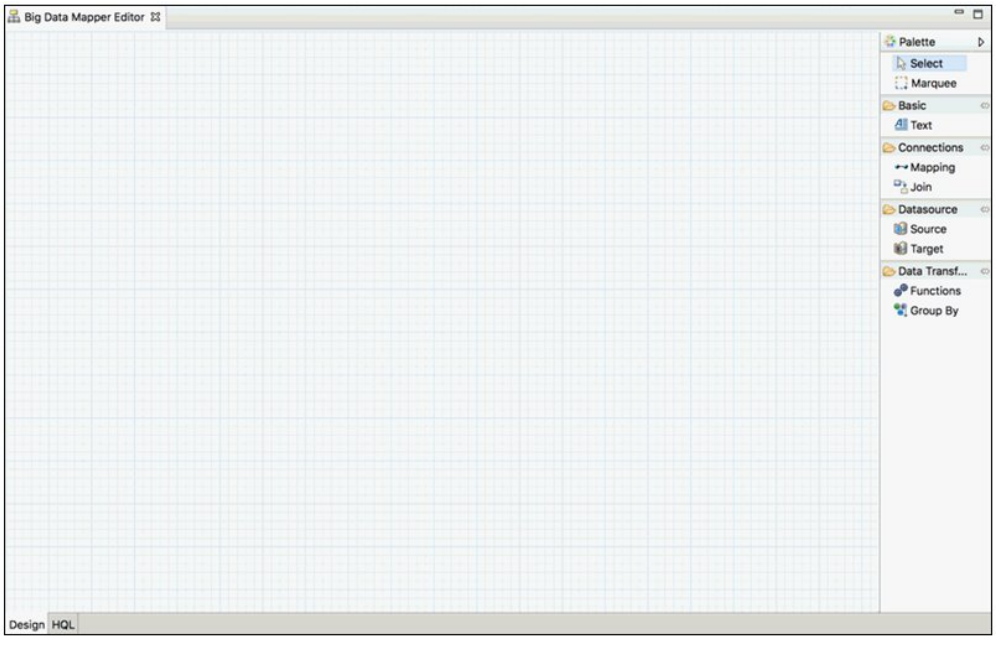
2. Type mapper in the field to filter the selection, select Big Data Mapper, and then click Next.

The New Mapper dialog opens, as shown in the following image.

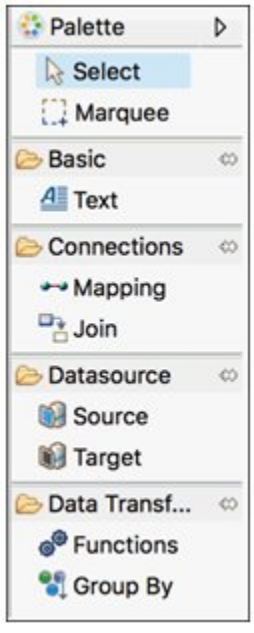


3. Enter a name for the mapper in the Name field (for example, *myBdiMapper*), select a Hive connection from the Connection Profile drop-down list, and then click *Finish*.

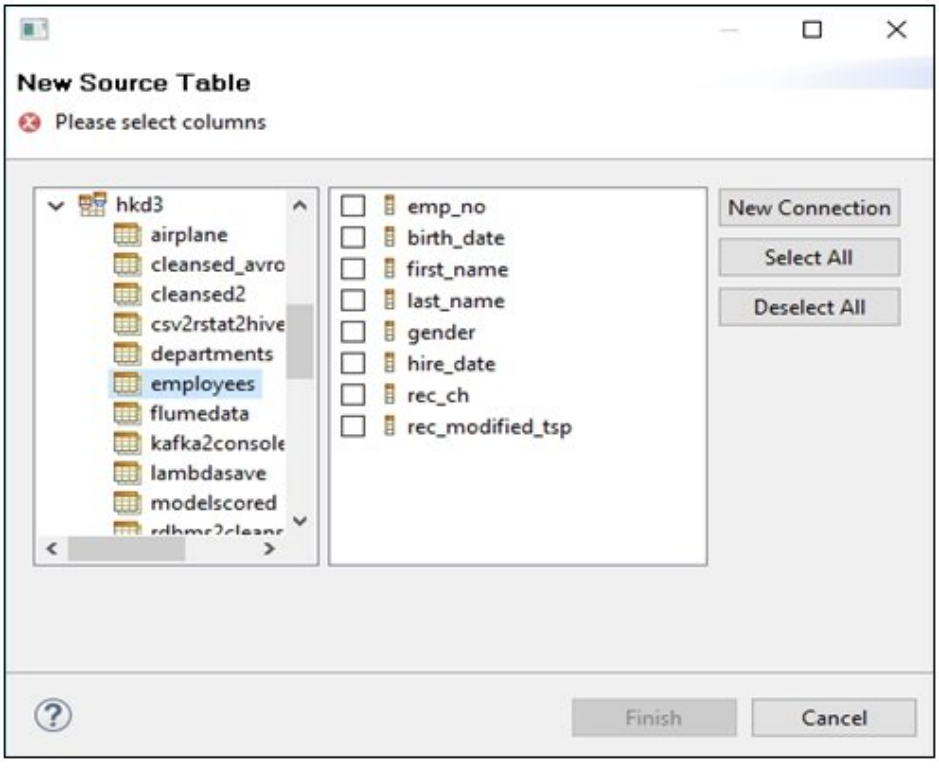
The Big Data Mapper Editor opens as a tab in the iBDI workspace, as shown in the following image.



4. From the Palette, which is located on the right pane of the workspace, click and drag the *Source* object (under the Datasource group) to the workspace.



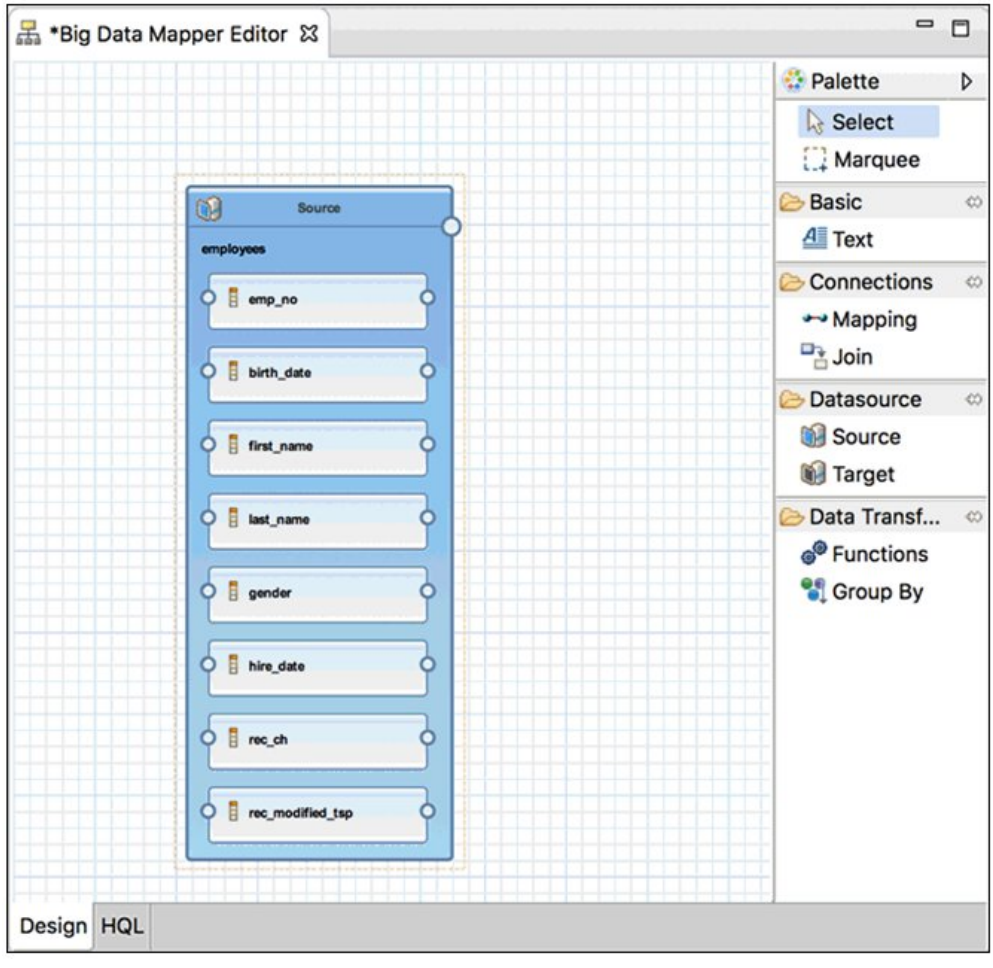
The New Source Table dialog opens, as shown in the following image.



The New Source Table dialog enables you to navigate and select the Hive table that you want to transform.

5. Click *Select All* and then click *Finish*.

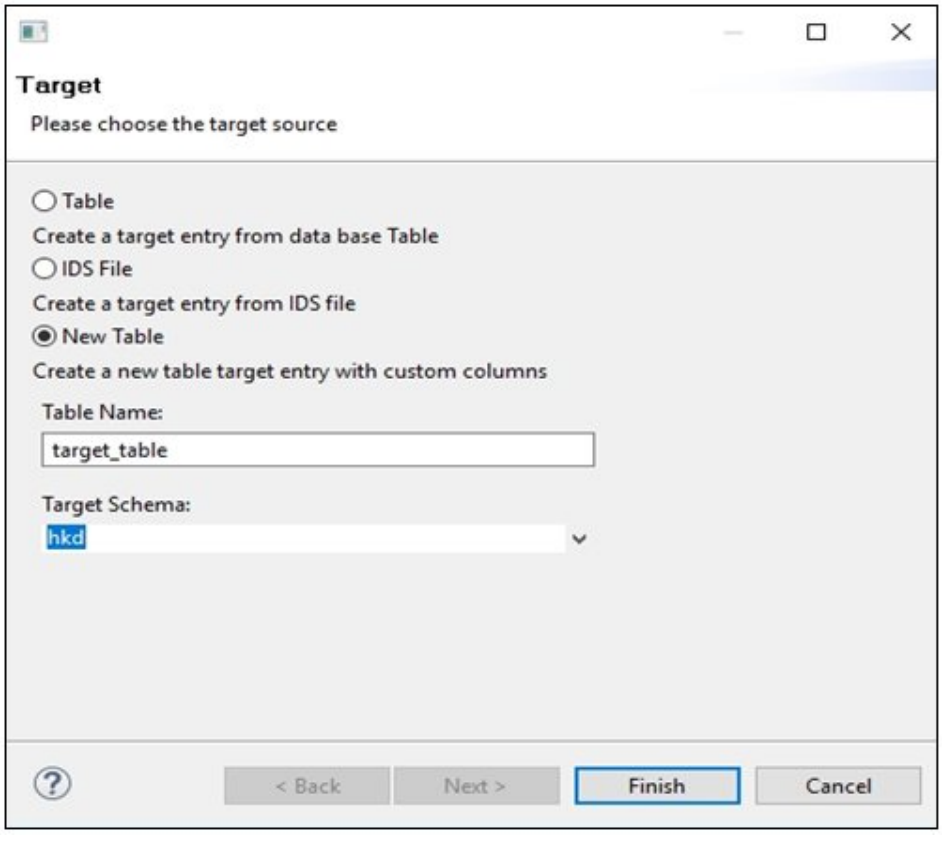
The Big Data Mapper Editor displays a graphical representation of your selected source tables, as shown in the following image.



Note: An asterisk character (*) in front of the Big Data Mapper Editor tab, indicates that a new change has been made to the mapping, which has not been saved. You can click the *Save* icon on the toolbar or use the *Ctrl*+S shortcut to save your work at any point.

6. From the Palette, click and drag the *Target* object (under the Datasource group) to the workspace.

The Target dialog opens, as shown in the following image.



The following table lists and describes the Table, IDS File, and New Table parameters.

Parameter	Description
Table	Creates the target entry in the selected schema and database with the selected columns. Selecting this option will open the New Source Table dialog. Selecting a table name in this dialog will write the contents of the mapper to the table.
IDS File	Import an Omni Input Document Specification (IDS) version 1 document and create columns in a new table based on the information in this document.

Parameter	Description
New Table	Create a table in the selected schema with the name provided. Only mapped columns from the Source will be written to the target.

7. Perform the following steps:

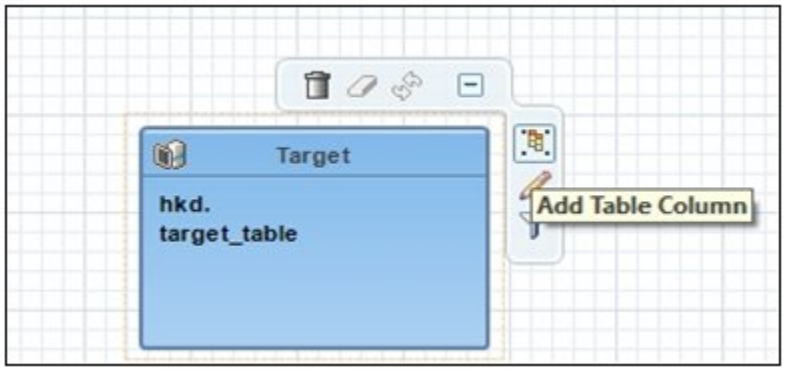
   a. Select *New Table*.

   b. Provide a table name (for example, *target_table*) in the Table Name field.

   c. From the Target Schema drop-down list, select a target schema into which the new table will reside. Here, the target schema is referring to a database.

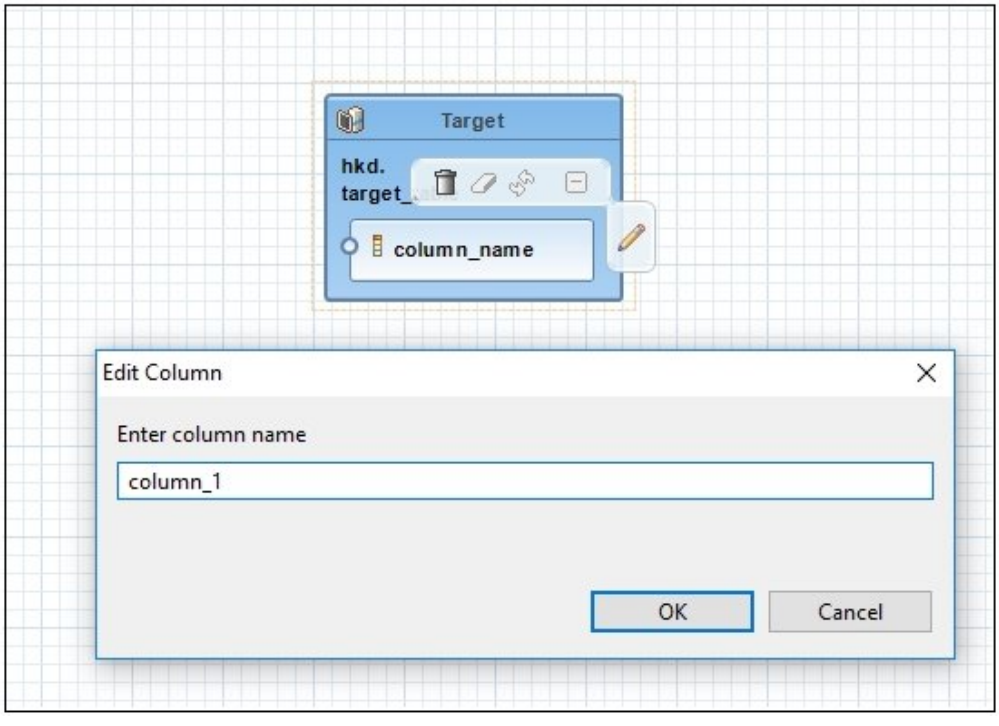
8. Click *Finish*.

   The Target object is added to the workspace with no columns, as shown in the following image.

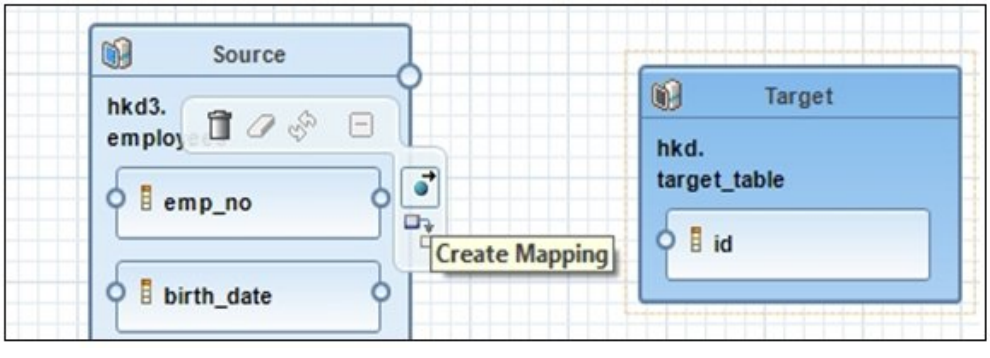


9. Add columns to the target table by hovering the pointer over the target table and clicking the *Add Table Column* icon.

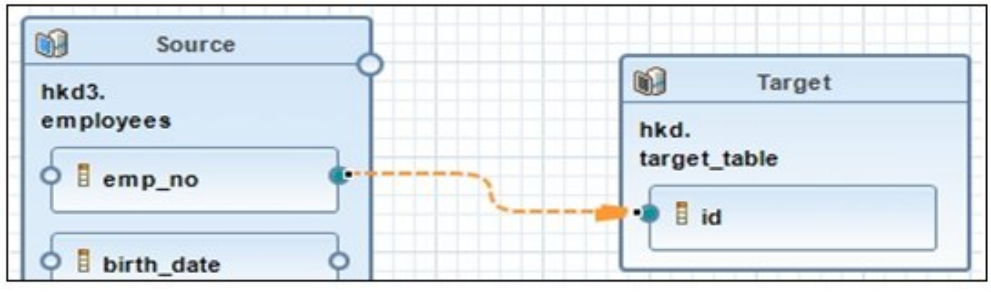
10. Edit the name of the column by clicking the *Edit Column* (pencil) icon when hovering over the added column, as shown in the following image.



11. Create a mapping join from the source table by hovering over a column in the source table and then clicking the *Create Mapping* icon, as shown in the following image.



12. Click the blue ball and arrow icon, drag the connection, and release it over the column created in the target table to form the connection between the source and target table, as shown in the following image.



13. Continue this process of creating a target column and creating a mapping connection from the source table.

14. To view the corresponding HiveQL (HQL) that is generated, click the *HQL* tab to view the HQL that Beeline or Spark will execute.

    For example:

```
CREATE DATABASE IF NOT EXISTS hkd;
CREATE TABLE IF NOT EXISTS hkd.target_table (
  id STRING)
STORED AS AVRO;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.exec.parallel=false;
--SET hive.exec.max.dynamic.partitions=100000;
--SET hive.exec.max.dynamic.partitions.pernode=100000;
create database if not exists hkd;
INSERT INTO TABLE hkd.target_table
SELECT
  T1.emp_no C1
FROM hkd3.employees T1
;
```

Procedure: How to Execute a Mapping Configuration

Once you have created a mapping configuration, a mapping file with a .iwmapper extension is added to your *Mappings* directory in your iWay Big Data Integrator (iBDI) project.

Note: When executing a mapping configuration in the Run Configurations facility, use the *iWay Big Data Integrator Build* option for immediate execution. Use the *iWay Big Data Integrator Publish* option for batch usage.

For more information, see *Defining Run Configurations* on page 147.

Chapter 11

Creating a Wrangler Configuration

This chapter introduces wranglers and describes how to create a wrangler configuration in iWay Big Data Integrator (iBDI).

In this chapter:

- Configuring a Connection to a Hadoop Distributed File System Server
- Creating a Wrangler Configuration

Configuring a Connection to a Hadoop Distributed File System Server

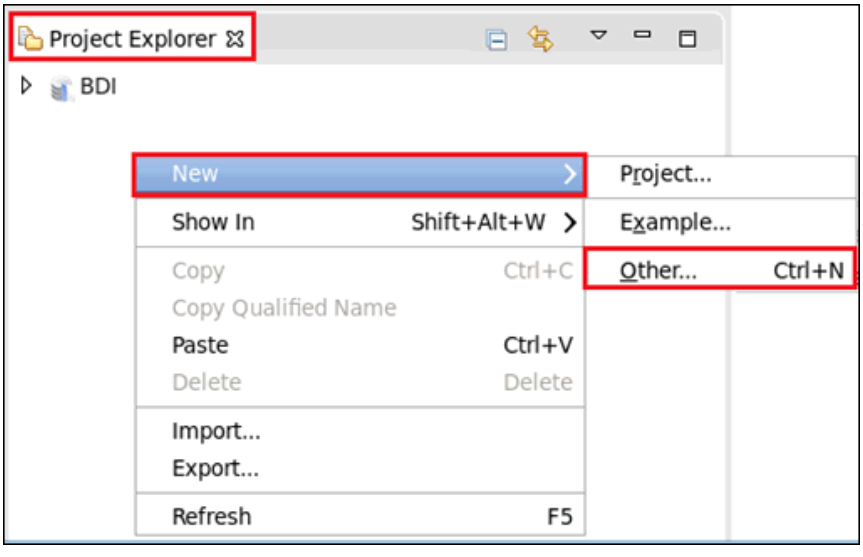
The wrangler is a bridge between unstructured Hadoop Distributed File System (HDFS) data and structured or defined data sources. Before you create a wrangler configuration, you must ensure that a connection a HDFS server is available.

HDFS is a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.

Note: Wranglers are a non-Pipeline type of data configuration.

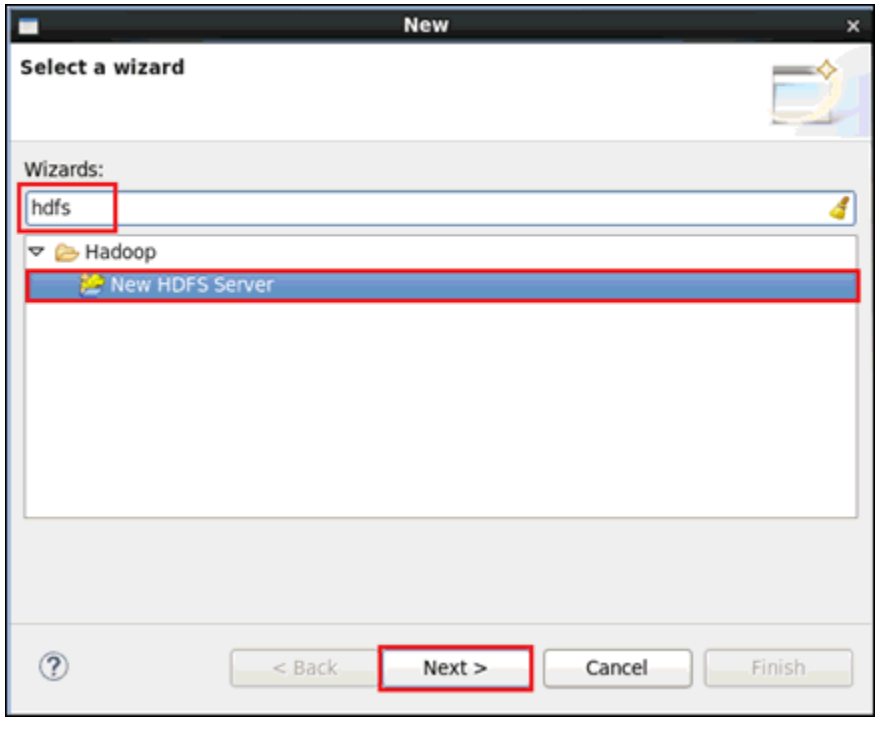
Procedure: How to Configure a Connection to a HDFS Server

1. Right-click anywhere within the Project Explorer tab, select *New*, and then click *Other* from the context menu, as shown in the following image.



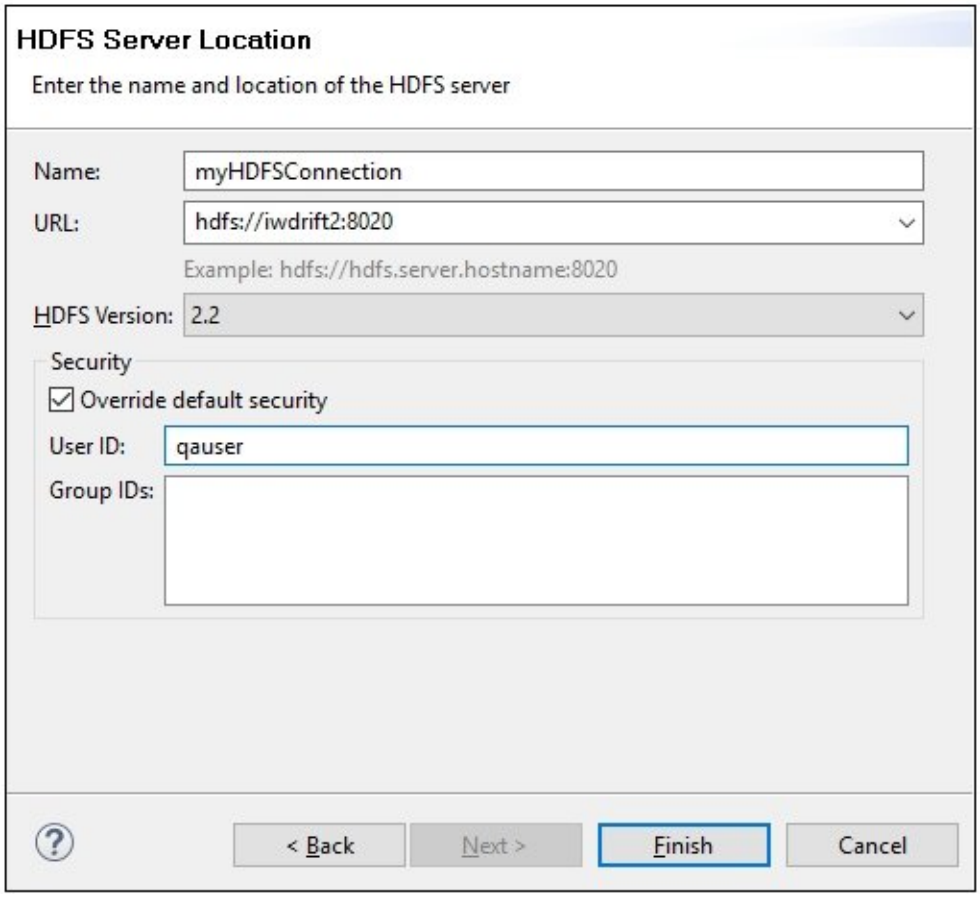
You can also click the following icon on the toolbar:

The New dialog opens, as shown in the following image.



2. Type *hdfs* in the field to filter the selection, select *New HDFS Server*, and then click *Next*.

The HDFS Server Location pane opens, as shown in the following image.



3. Specify the following information:

   - **Name:** Enter a name for the HDFS connection.
   - **URL:** Enter an HDFS connection URL with a port number. This is usually the URL of the system in the cluster running the Hadoop server.
   - **HDFS Version:** Select a HDFS version from the drop-down list.
   - **Security:** Enter an explicit user ID, group IDs, or leave blank for the default security implementation.

     For more information on user identity and usage, see the *Hadoop Distributed File System Permissions Guide*.

4. Click *Finish*.

   Your new connection to a HDFS server is listed as a new node in the Project Explorer tab.

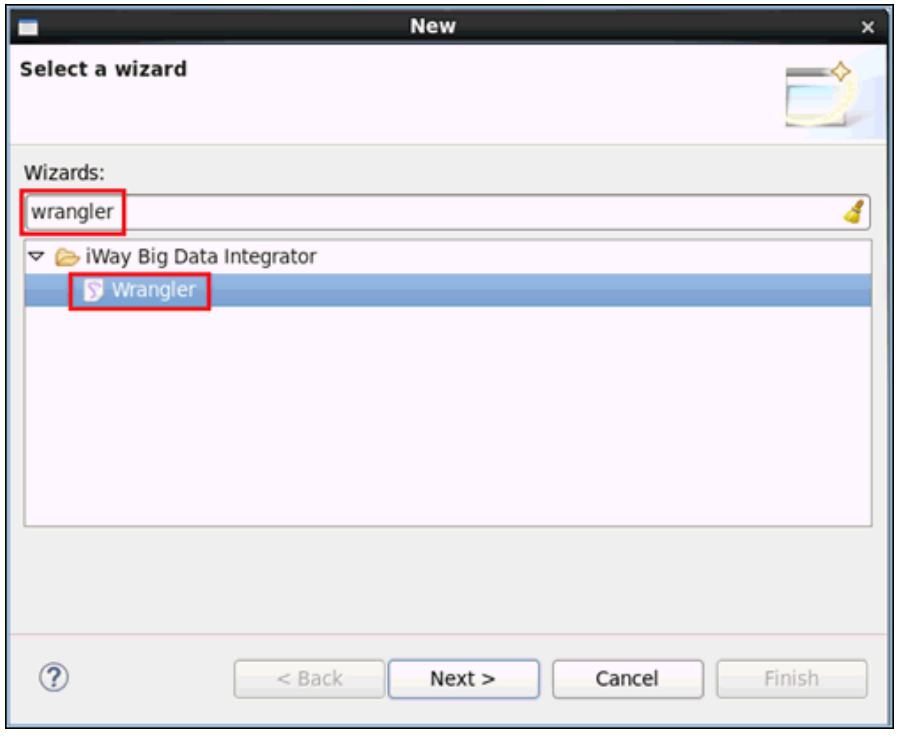
Creating a Wrangler Configuration

This section describes how to create a wrangler configuration in iWay Big Data Integrator (iBDI). The wrangler provides structure to unstructured data by assigning a schema to the data. The wrangler will conform the data to the schema.

Procedure: How to Create a Wrangler Configuration

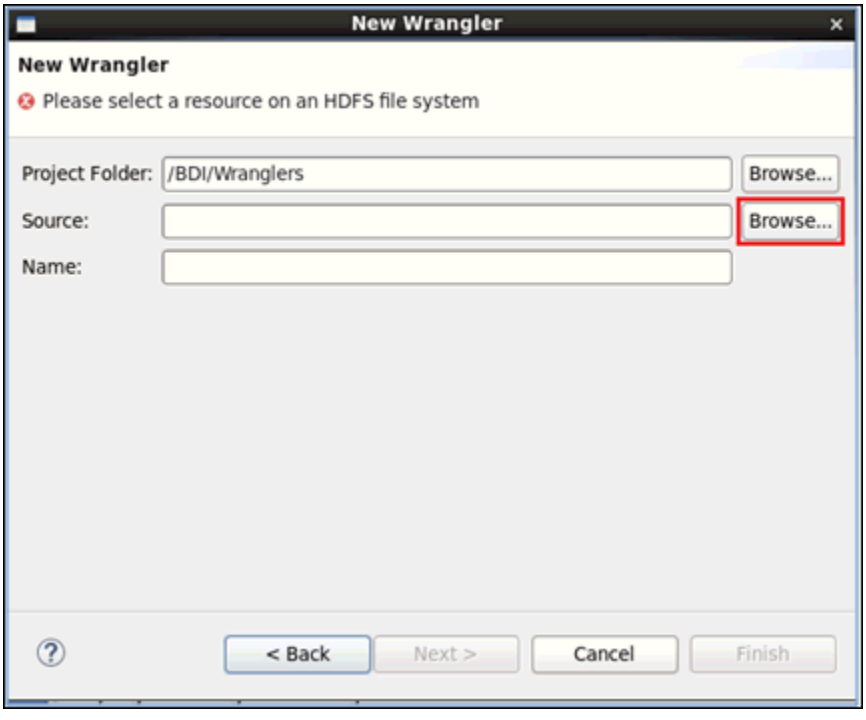
1. Expand an available iBDI project node in the Project Explorer tab, right-click the *Wranglers* folder, select *New*, and then click *Other* from the context menu.

   The New dialog opens, as shown in the following image.



2. Type *wrangler* in the Wizards field to filter the selection, select *Wrangler*, and then click *Next*.

The New Wrangler dialog opens, as shown in the following image.



The Project Folder field is automatically populated with a folder path, which you can modify as required.

3. Click *Browse* to the right of the Source field.

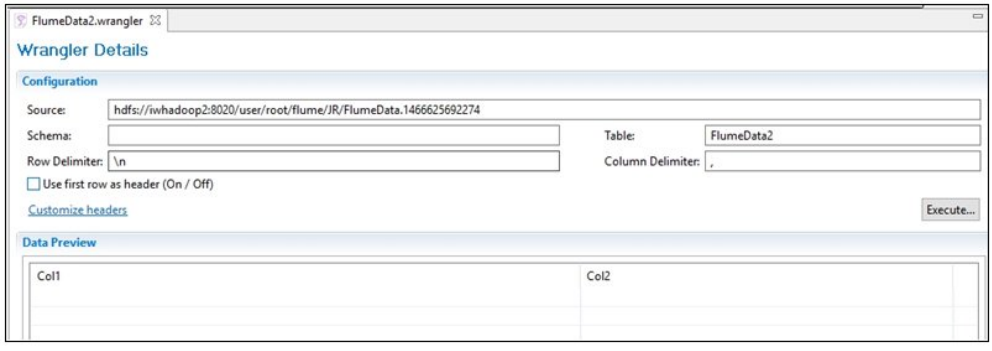
   The Select a Source dialog opens.

4. Navigate to the file that will be used by the wrangler you are configuring, select the file, and then click *OK*.

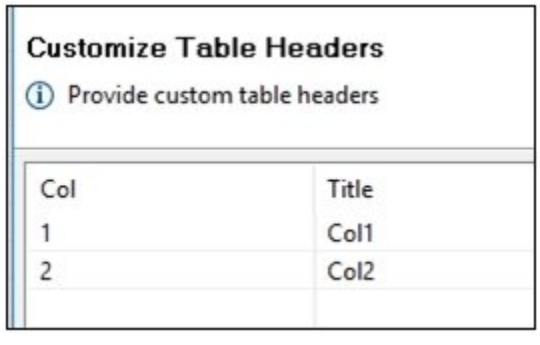
   You are returned to the New Wrangler dialog where the Source field and Name field are now populated.

5. Click *Finish*.

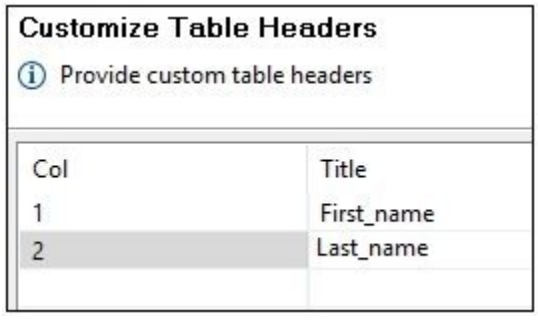
The wrangler opens as a new tab (for example, *FlumeData2.wrangler*) in the iBDI workspace, as shown in the following image.



6. Click *Customize headers* to modify the Header text.

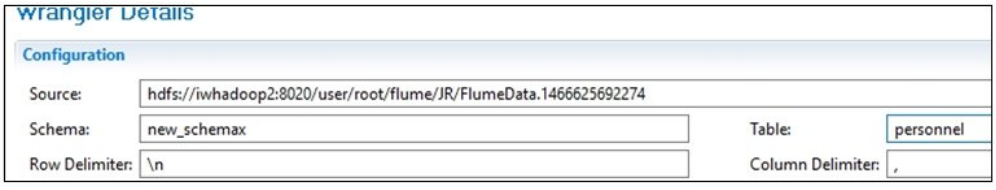


7. Click the title in the cell to change the name.



8. Click *OK* at the bottom of the dialog to commit the changes to the column headers.

9. In the Wrangler Details section, enter a schema name in Hive or RDBMS where the table will be created. This must be a currently connected data source.



10. In the Table field, enter the new table name to be created.
11. Enter or modify the Row Delimiter and Column Delimiter values.
12. Click *Execute* to write the changes to the database under the table and schema names that were specified.

    The console pane in the lower part of the iBDI user interface displays the results. For example:

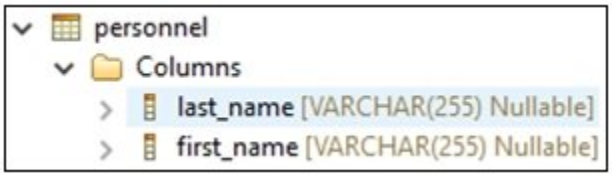
```
12/09/2016 03:30:43.590 [INFO] Executing statement: CREATE
DATABASE IF NOT EXISTS new_schemax
12/09/2016 03:30:43.610 [INFO] Executing statement: CREATE EXTERNAL
TABLE new_schemax.personnel ( First_name STRING , Last_name STRING ) ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'
LOCATION '/user/root/flume/JR'
12/09/2016 03:30:43.707 [INFO] Wrangler execution was completed
successfully.
```

13. Perform the following steps to verify the results:
    a. Return to the Data Source Explorer and browse to the schema and table that you specified in the previous steps (Step 9 and Step 10).

       The new table should be available.

    b. Right-click and select *Refresh* from the context menu or press *F5* to view the updated data.

       Note: If the result set is very large, this process may take several minutes to complete and appear under the menu, as shown in the following image.



Chapter 12

Defining Run Configurations

This chapter describes how to define run configurations using iWay Big Data Integrator (iBDI).

In this chapter:

- iBDI Runtime Configuration Considerations
- Defining Run Configurations for a Pipeline
- Defining Run Configurations for a Sqoop or a Flume
- Defining Run Configurations for a Spark Deployment

iBDI Runtime Configuration Considerations

There are three options in the Run Configurations dialog that you must consider when creating your runtime configuration.

The following two options are available for single action operations (for example, Sqoop, Flume, or Mapping):

- *iWay Big Data Integrator Deploy*
- *iWay Big Data Integrator Publish*

The following option is only available for Pipelines:

- *iWay Big Data Integrator Pipeline*

When you right-click *iWay Big Data Integrator Pipeline* in the Run Configurations dialog, you can select *Deploy* or *Publish* from the context menu that is displayed.

All of these options securely create the runtime configuration on the selected Edge node and copy all of the files that are required to run the selected iBDI process.

The following guidelines must be considered when choosing a Deploy or Publish option:

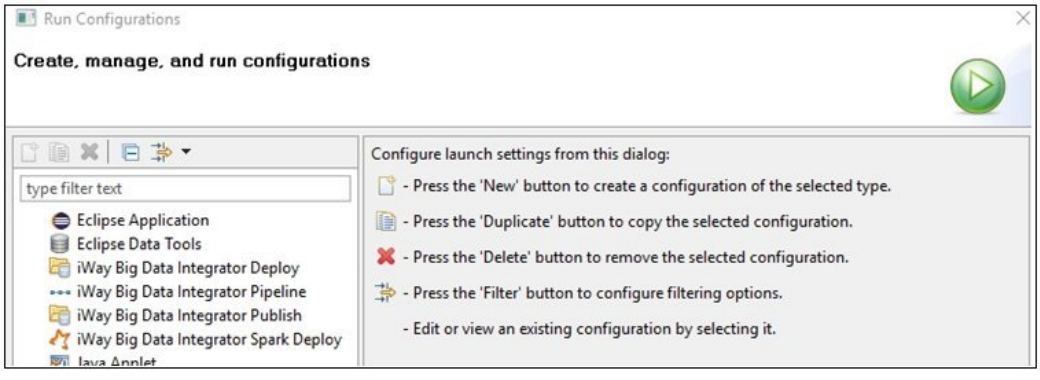
- **Deploy**. Used for small, interactive operations where the amount of data is not excessive. The trace log is written directly in the Console tab of the iBDI workspace area.
- **Publish**. Used for jobs that will be configured as a service on the Edge node, or for processes that run streams or listen for events (for example, Flume or Kafka).

If an event listener is attempted to run in Deploy mode, then it will try to start a process on the local client JVM in disconnected mode. This process can only be stopped by closing the iBDI client workbench, which stops the local JVM.

Defining Run Configurations for a Pipeline

Once a pipeline has been created and saved, right-click the pipeline project in the Project Explorer, select *Run As* from the context and then click *Run Configurations*. From the main menu, you can also select *Run* and then click *Run Configurations*.

The Run Configurations dialog opens, as shown in the following image.



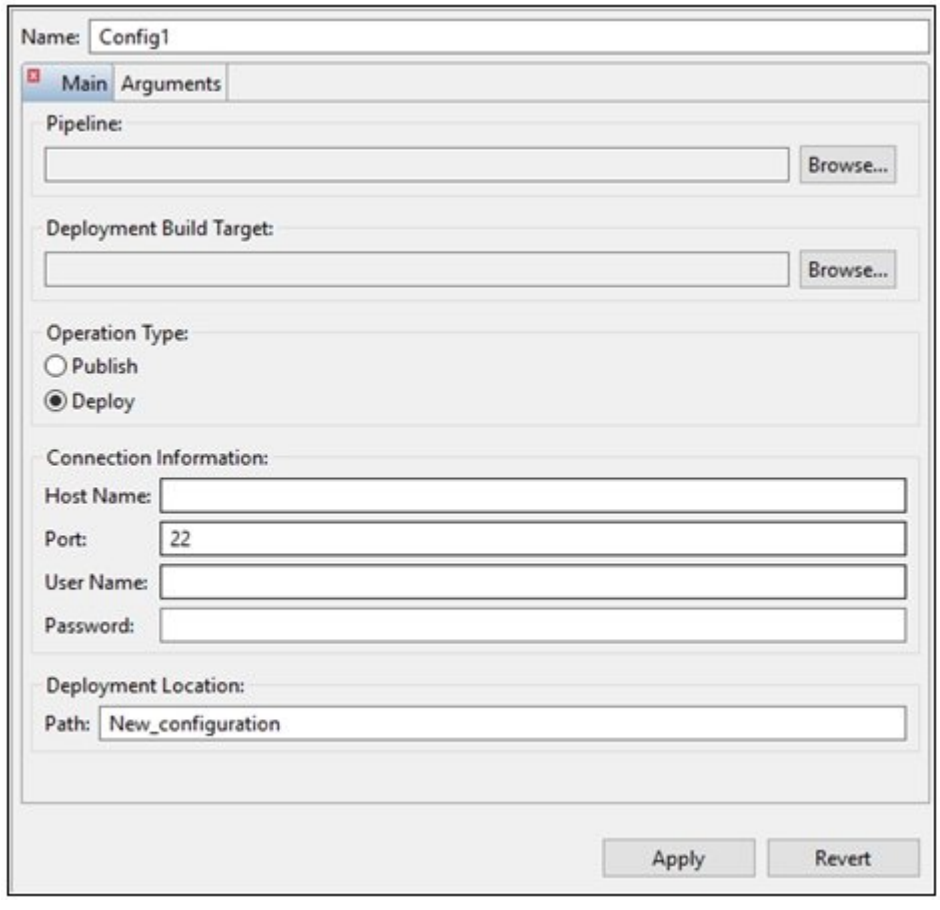
The Run Configurations dialog has two views. The left pane provides a list of applications. The right pane provides a series of instructions. You can resize the width of either pane accordingly by clicking and dragging the vertical bar that separates each pane.

Perform the following steps to define a run configuration for a pipeline:

1. Right-click *iWay Big Data Integrator Pipeline* in the left pane of the Run Configurations dialog and select *New* from the context menu.

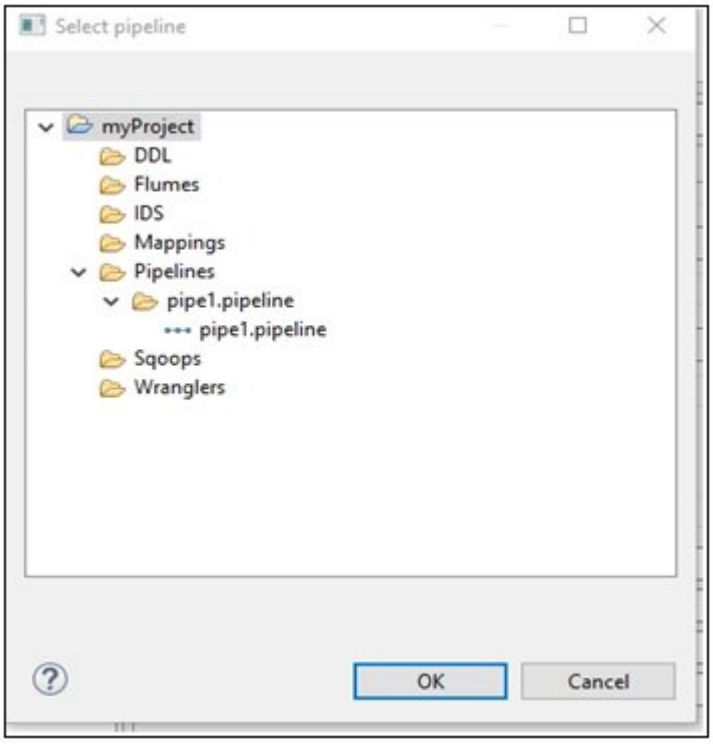
   Note: If an existing run configuration is available, you can right-click that instance and select *Duplicate* from the context menu. You can then modify only those parameter values that are required without having to configure all of the parameters.

A new run/build configuration pane opens, as shown in the following image.



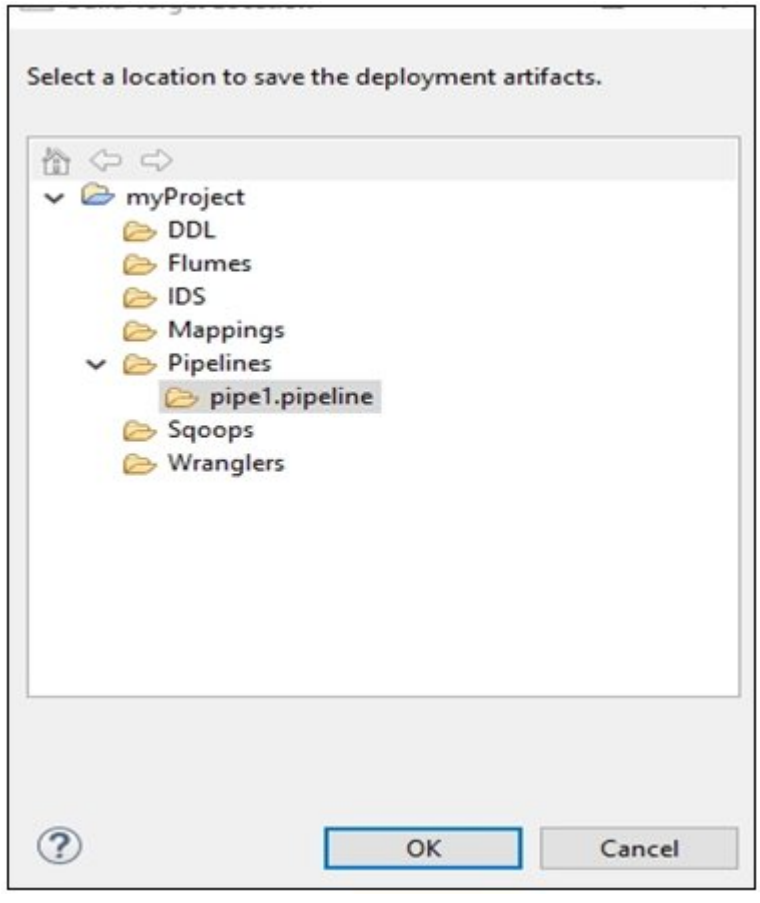
2. Enter a descriptive name for the new pipeline run configuration in the Name field. Avoid using dashes or slash characters in the name.
3. Click *Browse* to the right of the Pipeline field.

The Select pipeline dialog opens, as shown in the following image.



4. Expand the main project node, the *Pipelines* folder, the subfolder that corresponds to your pipeline project (for example, *pipe1.pipeline*), and select the pipeline project (for example, *pipe1.pipeline*).
5. Click *OK*.
6. Click *Browse* to the right of the Deployment Build Target field.

The Build Target Location dialog opens, as shown in the following image.



7. Expand the main project node, the *Pipelines* folder, and the subfolder that corresponds to your pipeline project (for example, *pipe1.pipeline*).
8. Click *OK*.
9. Select one of the following operation types:

   - **Publish.** Copy to the edge node.
   - **Deploy.** Copy to the edge node and run.

   An object is built or published when it will reside on the edge node and run through CRON jobs or deployment scripts. An object is deployed when it will interact with the console or produce interactive output.

10. Enter the following connection information:

    - Host Name
    - Port

      Note: Port 22 is the default for SSH, but is not the only port value that can be used.

    - User Name
    - Password

    The connection to the server will be secured through SSH.

11. In the Deployment Location field, enter the path to the iBDI provision location where the run time object will be deployed.
12. Click *Apply* to save the information that you specified.

    To clear all of the information that you specified in this dialog, click *Revert*.

13. Click *Run* to run the pipeline configuration.

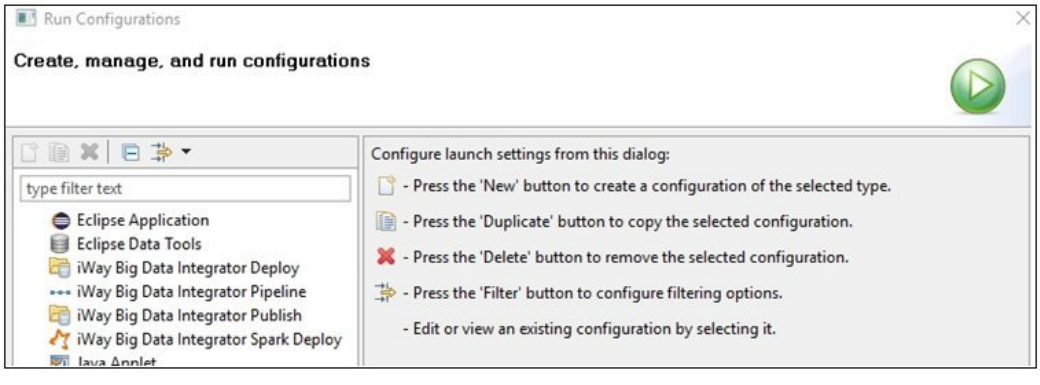
    The Console pane opens and displays the results for this deployment.

Defining Run Configurations for a Sqoop or a Flume

This section describes how to define run configurations for non-Spark pipeline types (for example, a Sqoop or a Flume).

Once a Sqoop or Flume configuration has been created and saved, right-click the Sqoop or Flume project in the Project Explorer, select *Run As* from the context and then click *Run Configurations*. From the main menu, you can also select *Run* and then click *Run Configurations*.

The Run Configurations dialog opens, as shown in the following image.



The Run Configurations dialog has two views. The left pane provides a list of applications. The right pane provides a series of instructions. You can resize the width of either pane accordingly by clicking and dragging the vertical bar that separates each pane.

Perform the following steps to define a run configuration for a Sqoop or Flume:

1. Right-click *iWay Big Data Integrator Deploy* or *iWay Big Data Integrator Publish* in the left pane of the Run Configurations dialog and select *New* from the context menu.

   - Selecting *iWay Big Data Integrator Deploy* deploys to the edge node and runs.
   - Selecting *iWay Big Data Integrator Publish* only publishes to the edge node.

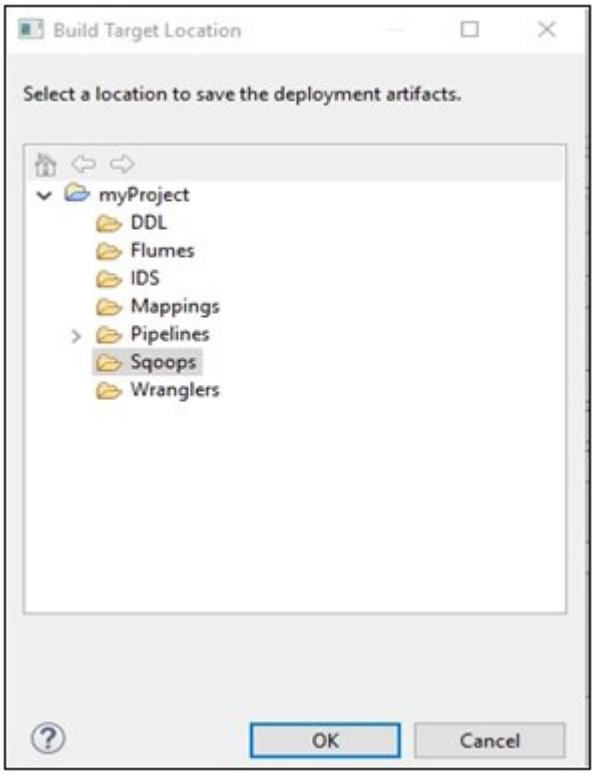
     Note: When defining a run configuration for a Flume, you must select the *iWay Big Data Integrator Publish* option from the context menu.

   Tip: If an existing run configuration is available, you can right-click that instance and select *Duplicate* from the context menu. You can then modify only those parameter values that are required without having to configure all of the parameters.

   A new run/build configuration pane opens.

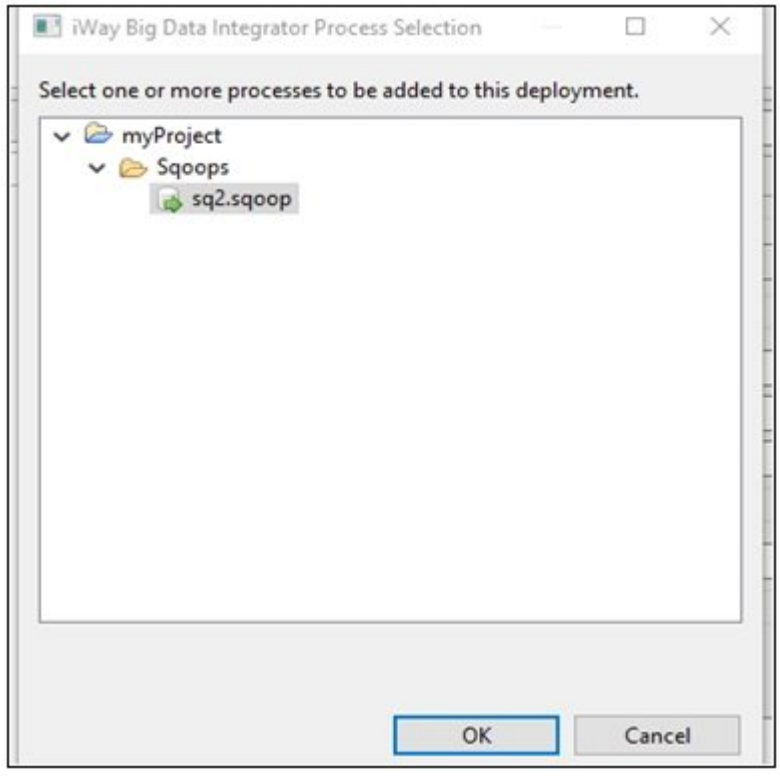
2. Enter a descriptive name for the new Sqoop or Flume run configuration in the Name field. Avoid using dashes or slash characters in the name.
3. Click *Browse* to the right of the Deployment Build Target field.

The Build Target Location dialog opens, as shown in the following image.



4. Expand the main project node and then the *Sqoops* or *Flumes* subfolder.
5. Click *OK*.
6. Enter the following connection information:
   - ❑ Host Name
   - ❑ User Name
   - ❑ Password
7. In the Deployment Location field, enter the path to the iBDI provision location where the run time object will be deployed.

8. To specify the value for the iWay Big Data Integrator Processes area, click the green plus sign icon (+), which opens the iWay Big Data Integrator Processes Selection dialog, as shown in the following image.



   Expand the main project node, *Sqoops* or *Flumes*, and then select the Sqoop or Flume project (for example, *sq2.sqoop*).

   Note: More than one process can be added in one single deploy or publish operation.

9. Click *OK*.
10. Click *Apply* to save the information that you specified.

    To clear all of the information that you specified in this dialog, click *Revert*.

11. Click *Run* to run the Sqoop or Flume configuration.

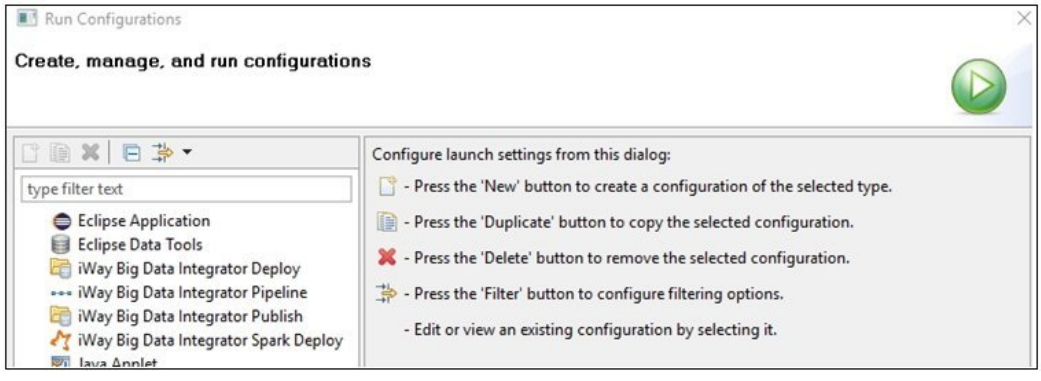
    The Console pane opens and displays the results for this deployment.

Defining Run Configurations for a Spark Deployment

Custom Spark code can be run with iWay Big Data Integrator (iBDI) using the *iWay Big Data Integrator Spark Deploy* option.

Perform the following steps to define a run configuration for a Spark deployment:

1. Right-click *iWay Big Data Integrator Spark Deploy* in the left pane of the Run Configurations dialog and select *New* from the context menu.



   Note: If an existing run configuration is available, you can right-click that instance and select *Duplicate* from the context menu. You can then modify only those parameter values that are required without having to configure all of the parameters.

   A new run/build configuration pane opens.

2. Enter a descriptive name for the new Spark deployment run configuration in the Name field. Avoid using dashes or slash characters in the name.
3. Click *Browse*.

   Navigate to the location of a Scala source file or Spark .jar file.

4. Click *OK*.
5. Enter the following connection information:

   - Host Name
   - User Name
   - Password

6. In the Deployment Location field, enter the path to the iBDI provision location where the run time object will be deployed.
7. Click *Apply* to save the information that you specified.

   To clear all of the information that you specified in this dialog, click *Revert*.

8. Click *Run* to run the Spark deployment.

   The Console pane opens and displays the results for this deployment.

Chapter 13

Pipeline Parameters Reference

This appendix provides a reference for the available pipeline parameters.

In this chapter:

- Pipeline Source Type Parameters
- Pipeline Compute Type Parameters
- Pipeline Target Types
- Data Format File Types
- Streaming Considerations

Pipeline Source Type Parameters

This section provides a reference for the available pipeline Source type parameters.

HDFS

- **File Type** (required)

  Select one of the following as the file type:

  - AVRO
  - CSV
  - JSON
  - ORC
  - PARQUET
  - TEXT
  - XML

- **Header**

  For CSV, writes a header in the first line if checked.

- **Infer Schema**

  For CSV, will attempt to infer a schema from the file.

- **Path** (required)

  Enter the path to the HDFS file using the UNIX file format specification, either relative or absolute. The *audit.csv* file is relative to the user. For example, the */user/myuser/* folder if the user is *myuser*. If */tmp/audit.csv* is searched relative to the root folder regardless of the user name.

Hive

The schemas from the catalog of the Hive Data Source created through the Data Source Explorer are available as input sources to the Transformer. Click *Add Transform* to open the Transformer palette. For more information, see *Understanding the Transformer* on page 99.

HQL

- **Data Source** (required)

  Select the Hive data source from the drop-down list.

- **SQL** (required)

  Enter the Hive Query Language SQL expression to retrieve the information. For more information, see the *Select* section in the *Apache Hive Language* manual.

Stream

- **Data Format Type** (required)

  Select one of the following as the data format type:

  - AVRO
  - CSV
  - JSON
  - ORC
  - PARQUET
  - TEXT
  - XML

- **Host Name** (required)

  The streaming host name.

- **Port** (required)

  Port from which events are streamed.

- **Schema**

  A schema in Avro format that describes the incoming data.

- **Union Stream Count**

  Combines the specified number of streams into a single stream.

Kafka

- **Batch Duration**

  The time interval to divide stream data into batches in milliseconds.

- **Consumer Group** (required)

  The Kafka Consumer group that the pipeline will use to consume a topic.

- **Data Format Type** (required)

  Select one of the following as the data format type:

  - AVRO
  - CSV
  - JSON
  - ORC
  - PARQUET
  - TEXT
  - XML

- **Schema**

  A schema in Avro format that describes the incoming data.

- **Topic** (required)

  The name of the Kafka Topic to consume.

- **Union Stream Count**

  Combines the specified number of streams into a single stream.

- **Zookeeper Quorum** (required)

  A list of host and port pairs in URI format that form the Kafka cluster.

Kafka Direct

- **Batch Duration**

  The time interval to divide stream data into batches in milliseconds.

- **Consumer Group** (required)

  The Kafka Consumer group that the pipeline will use to consume a topic.

- **Data Format Type** (required)

  Select one of the following as the data format type:

  - AVRO
  - CSV
  - JSON
  - ORC
  - PARQUET
  - TEXT
  - XML

- **Metadata Broker List**

  A list of Kafka brokers that host the metadata in *host:port* format.

- **Schema**

  A schema in Avro format that describes the incoming data.

- **Topic** (required)

  The name of the Kafka Topic to consume.

Nifi

- **Batch Duration**

  The time interval to divide stream data into batches in milliseconds.

- **Data Format Type** (required)

  Select one of the following as the data format type:

  - AVRO
  - CSV
  - JSON
  - ORC
  - PARQUET
  - TEXT
  - XML

- **Nifi URL**

  The URL that points to the location of the Nifi server. For example:

  `http://localhost:8080/nifi`

- **Port Name**

  Nifi output port name.

- **Schema**

  A schema in Avro format that describes the incoming data.

RDBMS

- **Data Source**

  Select the Data Source defined in the Data Source Explorer pane that contains the connection properties to the RDBMS source.

- **JDBC Fetch Size**

  Some drivers have a low JDBC fetch size that can be overridden by this property.

- **Lower Bound**

  Used against the partitioning column to calculate the increment or split for each parallel task.

- **Number of Partitions**

  Specify the number of partitions used for this request, which must be greater than 1.

- **Partition Column** (required)

  Enter a numeric column that will be used to partition the request.

- **SQL** (required)

  Enter an SQL statement to retrieve the data from the source through the JDBC driver.

- **Upper Bound**

  Used against the partitioning column to calculate the increment or split for each parallel task.

  The *Lower Bound*, *Upper Bound*, and *Number of Partitions* parameters can be used to tune the request for advanced users. If no values are specified for these fields, values are automatically calculated based on the request.

Pipeline Compute Type Parameters

This section provides a reference for the available pipeline Compute type parameters.

Cleanse

Enter the URL of the Omni data quality service (JSD). For example:

```
http://myserver.mycorp.com:port/file.json?jsd
```

The cleansed fields are appended to the input source as columns.

Coalesce

Returns a DataFrame with exactly the number of partitions provided. If reducing from 1000 to 100 partitions, each partition will claim 10 of the current partitions. No shuffle is performed.

Distinct

Returns a DataFrame that contains only the unique rows from the source. There are no parameters.

Drop

- **Column**

  Enter the name of the column to drop. No operation is performed if the column name is not present.

Filter

- **Condition Expression**

  Enter an expression that will be used to filter the rows. For example:

  ```
  age > 15
  ```

  The result is the filtered set.

Hive

Add, edit, or reset a Hive table transform. For more information, see *Understanding the Transformer* on page 99.

HQL

- **Data Source** (required)

  Select the Hive data source from the drop-down list.

- **SQL**

  Enter the Hive Query Language SQL expression to retrieve the information. For more information, see the *Select* topic in the *Apache Hive Language* manual.

RStat Model

- **Class Name**

  The RStat model class name. If this name is not entered, then the first class in the .jar file will be used as the class name.

- **Jar File**

  The Java .jar archive containing the model exported by RStat. The models are static and do not *learn*.

Order By

Enter a comma separated list of columns. Returns a new DataFrame sorted by these columns.

Sample

- **Fraction**

- **Without Replacement**

Probability that each element is chosen. Fraction must be [0,1].

- **With Replacement**

  Expected number of times each element is chosen. Fraction must be greater than 0.

- **With Replacement**

  Checkbox {on | off}. Default is *off*.

  If set to *on*, then elements can be sampled multiple times (replaced when sampled out).

Save

Saves the data into Hive without stopping the pipeline.

- **Data Source** (required)

  Select the Hive data source from the drop-down list.

- **Data Format Type** (required)

  Select one of the following as the data format type:

  - AVRO
  - CSV
  - JSON
  - ORC
  - PARQUET
  - TEXT
  - XML

- **Mode**

  Overwrite an existing file or append to an existing file.

- **Schema** (required)

  Schema location in Hive where the table will be written.

- **Table Name** (required)

  Specify a name for the Hive table.

Select

Select an ordered list of columns from the DataFrame.

- **Columns**

  Enter columns that appear from the preceding source or compute node.

Show

Shows the first 20 records of a DataFrame. There are no parameters.

Pipeline Target Types

This section provides a reference for the available parameters that are related to the pipeline target types.

Console

Prints the results to the Console pane in iBDI. There are no parameters.

HDFS

Writes the DataFrame to HDFS.

- **Data Format Type** (required)

  Select one of the following as the data format type:

  - AVRO
  - CSV
  - JSON
  - ORC
  - PARQUET
  - TEXT
  - XML

- **HDFS Path** (required)

  Enter the path to the HDFS file using the UNIX file format specification, either relative or absolute. The *audit.csv* file is relative to the user. For example, the */user/myuser/* folder if the user is *myuser*. If */tmp/audit.csv* is searched relative to the root folder regardless of the user name.

- **Mode**

  Overwrite an existing file or append to an existing file.

HDFS CSV

Writes the DataFrame to HDFS in CSV format.

- **Compression Codec**

  No compression is used if no codec value is specified for this parameter. Enter a valid codec name (for example, bzip2,gzip, lz4, or snappy) to compress the file.

- CSV Delimiter

  Default is a comma (,) character. Enter a substitute character here, for example, a semicolon (;).

- **HDFS Path** (required)

  Enter the path to the HDFS file using the UNIX file format specification, either relative or absolute. The *audit.csv* file is relative to the user. For example, the */user/myuser/* folder if the user is *myuser*. If */tmp/audit.csv* is searched relative to the root folder regardless of the user name.

- **Mode**

  Overwrite an existing file or append to an existing file.

Kafka

Writes the result to a Kafka topic.

- **Kafka Brokers**

  Specify a list of Kafka brokers in *host:port* format, separated by comma (,) characters.

- **Topic**

  Enter the name of the Kafka topic to publish to.

RDBMS

Write the result to an RDBMS table.

- **Data Source**

Select the Data Source defined in the Data Source Explorer pane that contains the connection properties to the RDBMS source.

- **Mode**

  Overwrite an existing file or append to an existing file.

- **Table Name**

  Specify a table name that will be used to write the results.

Data Format File Types

This section lists and describes the supported data format file types in pipeline source type parameters, pipeline compute type parameters, and pipeline target types.

- **AVRO.** A data serialization system that relies on schemas to read and write data. Schemas are stored with the data when Avro data is serialized. Avro schemas are usually defined in JSON format.
- **CSV.** A text file format that uses a delimiter (most often a comma character) to separate fields. RFC 4180 contains the specification for this format. This format can be highly variable as the delimiter is often different by country or even company site.
- **JSON.** Based on a subset of the JavaScript programming language, it is a text format based on name/value pairs. ECMA-404 defines the JSON standard.
- **ORC.** A self typing, column oriented data store optimized for Hadoop. ORC is defined with compression, light weight indices, and contain metadata about the file. ORC supports the Hive data model and is a format that can be split.
- **PARQUET.** Columnar data start that uses record shredding and assembly for extremely efficient compression and encoding schemes. Stores full metadata at the end of files, for self documenting data files. Compression can be specified on a per column basis.
- **TEXT.** Flat text format requires a full data scan for every query. Data must be delimited for rows and columns, and fields that may contain the delimiter character must be escaped. Text may be very verbose and require compression that limits the parallelism of Hadoop or Spark. Usually does not contain an internal schema.
- **XML.** A structured text based file format that must be parsed usually with a schema to obtain the values contained in the file. XML files are usually uncompressed and have the schema stored separately from the data.

Streaming Considerations

This section provides notes and considerations that are related to streaming in iWay Big Data Integrator (iBDI).

Streaming Data Structure

Since the structure of streaming data is unknown, the user can provide an Avro schema to define it. Even if the data is not in Avro format, the metadata can be used to build a schema for the DataFrame. If the data format happens to be Avro, iBDI will use the schema at runtime to read the streaming data.

Defining a Structure Using Avro Schema

The Avro documentation contains the most concise explanation on defining a schema in JSON format. For more information, see the *Schema Declaration* topic in the *Apache Avro* documentation.

Data Types

Avro has built in primitive types to define data. For more information, see the *Primitive Types* topic in the *Apache Avro* documentation.

The following sample schema illustrates how iBDI will use an Avro schema to provide structure to the incoming stream, and search for values in the data.

```
{"namespace": "example.avro",
"type" : "record" ,
"name" : "User" ,
"fields" : [
{ "name" : "name" , "type" : "string" },
{ "name" : "favorite_number" , "type" : [ "int" , "null" ]},
{ "name" : "favorite_color" , "type" : [ "string" , "null" ]}
]
}
```

XML and JSON Data Streams

If the incoming stream data is XML, then iBDI will use the fields in the Avro schema to search for elements in the XML document. If the element is found, then its value will used as the value of that column in the DataFrame. JSON data works similarly in that the fields in the Avro schema are used to search for fields in the JSON object.

iBDI only supports flat JSON/XML objects. JSON/XML objects with complex, multi-level structures are not supported.

Nifi Streaming

The Apache Nifi wiki topic *Nifi Components* is a good place to start understanding the concepts behind Nifi. The developer documentation provides information on installing Apache Nifi. The HortonWorks community connection has an article on configuring HortonWorks specific Nifi with Spark Streaming (*Nifi+Spark:Feeding Data to Spark Streaming*).

Chapter 14

Configuring Kerberos Authentication

This appendix describes how to configure Kerberos authentication.

In this chapter:

- Creating the Java Authentication and Authorization Service Login Configuration File
- Configuring Kerberos Authentication on Windows
- Configuring Kerberos Authentication on Linux
- Configuring Kerberos Authentication on Mac OS X

Creating the Java Authentication and Authorization Service Login Configuration File

For all Kerberos authentication implementations (Windows, Linux, and MAC OS X), you must obtain a *jaas.conf*, *krb5.conf*, and a *keytab* file from your Hadoop system administrator.

You can create a Java Authentication and Authorization Service (JAAS) login configuration file (*jaas.conf*) by using the following sample for reference purposes:

```
com.sun.security.jgss.initiate {
    com.sun.security.auth.module.Krb5LoginModule required
    useKeyTab=true
    keytab="THE/PATH/TO/YOUR/KEYTAB"
    useTicketCache=true
    principal="USERNAME@HOST.DOMAIN.COM"
    debug=true;
};
```

Once your *jaas.conf* file is created, specify the following values in the *iWayBigDataIntegrator.ini* or *eclipse.ini* file:

```
## Java properties for Kerberized Hadoop

Djava.security.auth.login.config=/PATH/TO/YOUR/jaas.conf

Djavax.security.auth.useSubjectCredsOnly=false

Djava.security.krb5.realm=HOST.DOMAIN.COM

Djava.security.krb5.kdc=KDC.HOST.DOMAIN.COM
```

The *iWayBigDataIntegrator.ini* file is located in the root folder your iBDI installation.

For reference purposes, the following are sample JDBC URLs for Kerberos:

```
jdbc:hive2://localhost:10000/default;principal=hive/_HOST@HOST.DOMAIN.COM
```

```
jdbc:hive2://localhost:10000;principal=hive/_HOST@HOST.DOMAIN.COM;AuthMech=1;
KrbRealm=HOST.DOMAIN.COM;KrbHostFQDN=HOST.DOMAIN.COM;KrbServiceName=hive
```

```
jdbc:hive2://localhost:10000/default;principal=hive/_HOST@HOST.DOMAIN.COM;AuthMech=1;
KrbRealm=HOST.DOMAIN.COM;KrbHostFQDN=HOST.DOMAIN.COM;KrbServiceName=hive
```

Configuring Kerberos Authentication on Windows

You can configure your Kerberos setup so that you can use the MIT Kerberos Ticket Manager to get the Ticket Granting Ticket (TGT), or configure the setup so that you can use the driver to get the ticket directly from the Key Distribution Center (KDC). Also, if a client application obtains a Subject with a TGT, you can use that Subject to authenticate the connection.

Download and Install MIT Kerberos for Windows

This section describes how to download and Install MIT Kerberos for Windows.

Procedure: How to Download and Install MIT Kerberos for Windows

To download and install MIT Kerberos for Windows 4.0.1:

1. Download the appropriate Kerberos installer.
   - For a 64-bit computer, use the following download link from the MIT Kerberos website:

     http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-amd64.msi
   - For a 32-bit computer, use the following download link from the MIT Kerberos website:

     http://web.mit.edu/kerberos/dist/kfw/4.0/kfw-4.0.1-i386.msi
2. Run the installer by double-clicking the .msi file that you downloaded in step 1.
3. Follow the instructions in the installer to complete the installation process.
4. When the installation completes, click *Finish*.

Using MIT Kerberos Ticket Manager to Get Tickets

This section describes how to configure the KRB5CCNAME Environment Variable and get a Kerberos ticket. You must first set the KRB5CCNAME environment variable to your credential cache file.

Procedure: How to Set the KRB5CCNAME Environment Variable:

To set the KRB5CCNAME environment variable:

1. Click the *Start* button, right-click *Computer*, and then click *Properties*.
2. Click *Advanced System Settings*.
3. In the System Properties Dialog box of the Advanced tab, click *Environment Variables*.
4. In the Environment Variables dialog box, under the System variables list, click *New*.
5. In the Variable name field of the New System Variable dialog box, type *KRB5CCNAME*.
6. In the Variable Value field, type the path for your credential cache file.

   For example:

   ```
   C:\KerberosTickets.txt
   ```

7. Click *OK* to save the new variable and then ensure that the variable appears in the System Variables list.
8. Click *OK* to close the Environment Variables dialog box, and then click *OK* to close the System Properties dialog box.
9. Restart your computer.

Procedure: How to Get a Kerberos Ticket

To get a Kerberos ticket:

1. Click the *Start* button, then click All Programs, and click the *Kerberos for Windows (64-bit)* or *Kerberos for Windows (32-bit)* program group.
2. Click *MIT Kerberos Ticket Manager*.
3. In the MIT Kerberos Ticket Manager, click *Get Ticket*.
4. In the Get Ticket dialog box, type your principal name and password, and then click *OK*.

   If the authentication succeeds, then your ticket information appears in the MIT Kerberos Ticket Manager.

Using the Driver to Get Tickets

To enable the driver to get Ticket Granting Tickets (TGTs) directly, you must ensure that the KRB5CCNAME environment variable has not been set. This section describes how to delete the KRB5CCNAME environment variable.

Procedure: How to Delete the KRB5CCNAME Environment Variable

To delete the KRB5CCNAME environment variable:

1. Click the *Start* button, right-click *Computer*, and then click *Properties*.
2. Click *Advanced System Settings*.
3. In the System Properties dialog box, click the Advanced tab and then click *Environment Variables*.
4. In the Environment Variables dialog box, check if the KRB5CCNAME variable appears in the System variables list. If the variable appears in the list, then select the variable and click *Delete*.
5. Click *OK* to close the Environment Variables dialog box, and then click *OK* to close the System Properties dialog box.

Configuring the Kerberos Configuration File

This section describes how to configure the Kerberos configuration file.

Procedure: How to Set up the Kerberos Configuration File

To set up the Kerberos configuration file:

1. Create a standard krb5.ini file and place it in the C:\Windows directory.
2. Ensure that the KDC and Admin server specified in the krb5.ini file can be resolved from your terminal. If necessary, you can modify the following:

   ```
   C:\Windows\System32\drivers\etc\hosts
   ```

Configuring Kerberos Authentication on Linux

Depending on the Linux operating system being used, there are built in Kerberos packages, or packages may need to be installed. For more information, see the documentation for your Linux operating system:

- For Suse Linux, see the *Configuring a Kerberos Client with Yast* topic.
- For Ubuntu Linux, see the *Kerberos Linux Client* topic.
- For Centos/Red Hat, see the *Configuring a Kerberos 5 Client* topic.

In general, to obtain a Kerberos ticket, enter the following `kinit` command:

```
kinit -p user@REALM.TOP
```

where:

REALM

Is the unique domain name, which may be delineated by multiple intermediate domains and an endpoint domain.

TOP

Is the suffix COM, ORG, EDU, or other signifier.

After entering the `kinit` command, the operating system will prompt you for the principals password:

```
Password for myBDItest@MYREALM.TOP:
```

Specify the REALM password for the user, which may be different than the current user logon ID password.

The `kinit` command should return without an error message.

Enter the `klist` command to view the ticket:

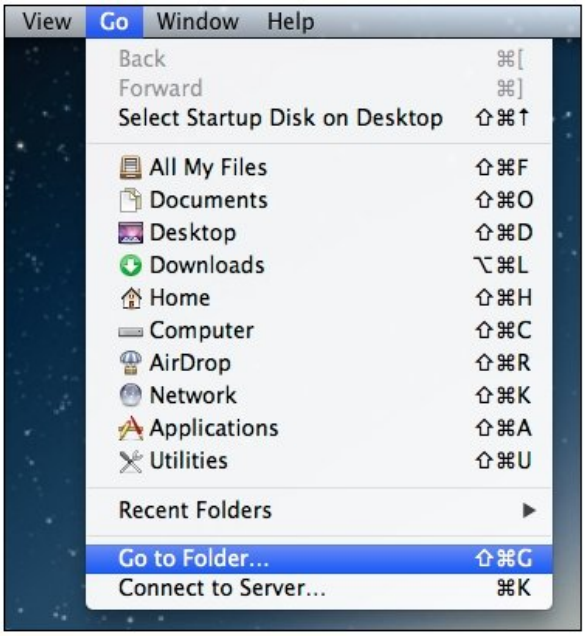
```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: myBDItest@MYREALM.TOP

Valid starting     Expires            Service principal
11/22/16 09:19:22  11/22/16 13:18:40  krbtgt/ MYREALM.TOP @ MYREALM.TOP
renew until 11/22/16 13:19:22
```

Configuring Kerberos Authentication on Mac OS X

Mac OS X has a Kerberos client installed with the operating system. The *Ticket Viewer* application provides a graphical user interface for obtaining Kerberos tickets.

Procedure: How to Configure Kerberos Authentication on Mac OS X

1. From the main menu at the top of the screen, select *Go*, then click *Go to Folder*, as shown in the following image.



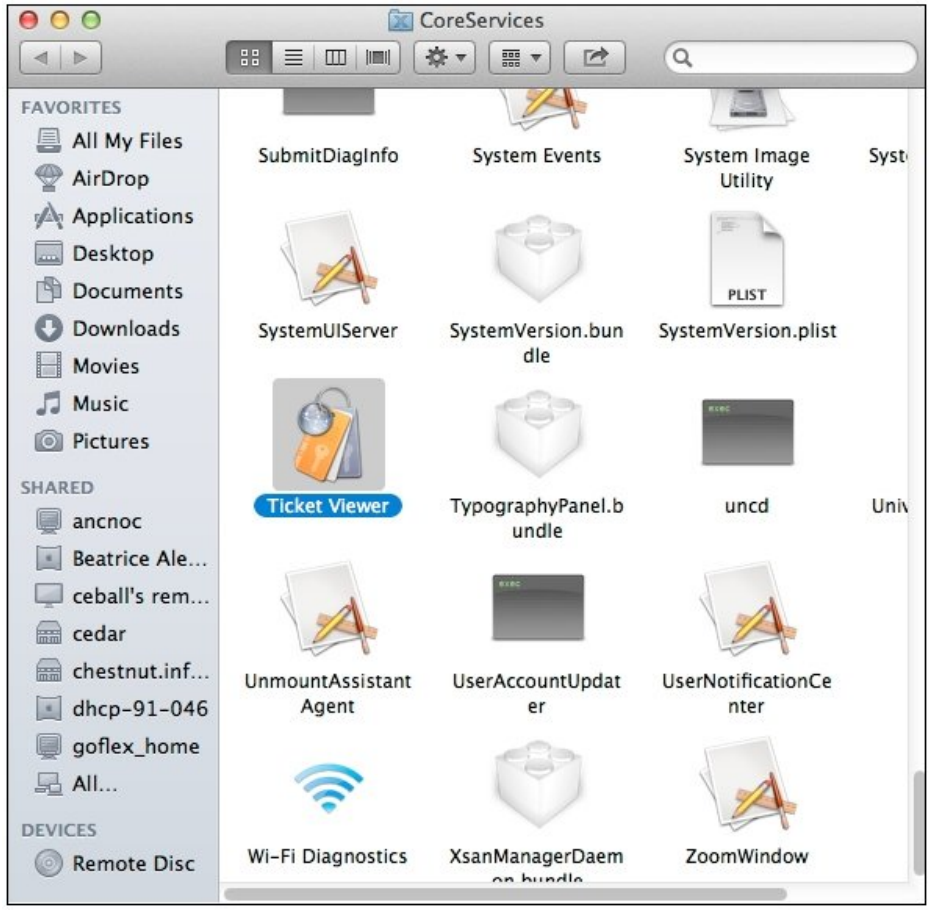
The Go to Folder dialog opens, as shown in the following image.

2. Specify the following path:

   ```
   /System/Library/CoreServices/
   ```

3. Click *Go*.

The CoreServices folder opens, as shown in the following image.

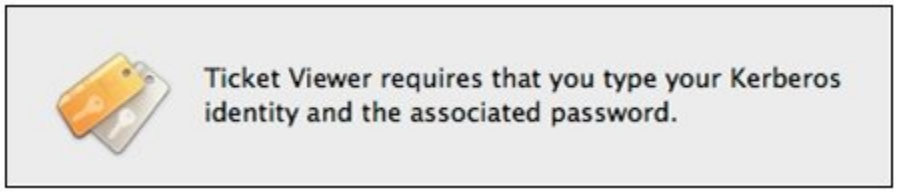


4. Click *Ticket Viewer*.

   The Ticket Viewer application opens, as shown in the following image.

5. Click *Add Identity*.

The following message is displayed:



6. Enter the assigned Kerberos principal name using the following format:

   `user@REALM.TOP`

   where:

   REALM

   Is the Kerberos realm, such as MYREALM.

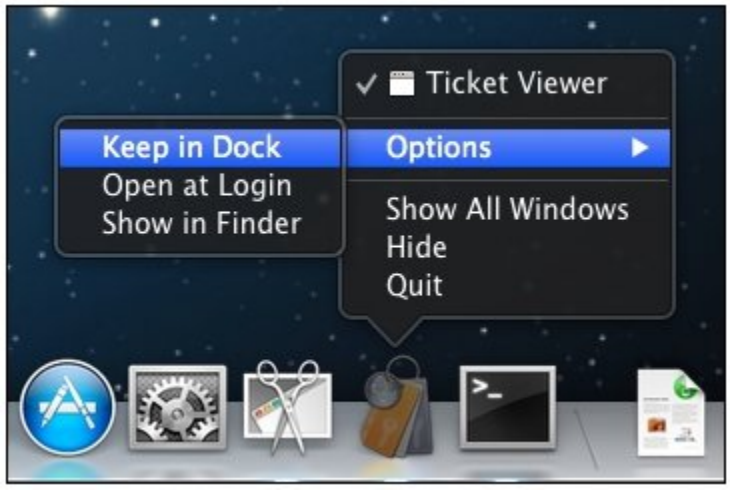
   TOP

   Is the top level identifier, such as COM or ORG.

7. Click *Continue* to obtain the Kerberos ticket.

8. Click *Set as Default* to install the ticket.

9. Set the Ticket Viewer application in the dock for easy access, as shown in the following image.



Glossary

This appendix provides a reference for common terms that are used in Big Data discussions and iWay Big Data Integrator (iBDI).

Avro	A Row-wise data serialization file format used in Hadoop. Avro files can be queried with Hive. Use Avro if you intend to consume most of the columns in the row.
Change Data Capture	Techniques and solutions for managing incremental data loads.
Flume	Flume is used to import data logs into HDFS while it is generated. For example, it imports web servers, application servers, and click streams, while simultaneously using Hadoop to clean up the data, aggregate it, and store it for long-term use. Data can then be outputted to the BI and batch reports or *intercepted* and altered before writing it to disk. Metadata (for example, rows and columns) can be applied to streaming source data to provide a look and feel of a database.
Hadoop	An open-source framework in Java for distributed storage and processing of vary large data sets on computer clusters built from commodity hardware.
HDFS	Hadoop Distributed File System (HDFS) is composed of one NameNode and multiple data nodes which contain blocks of files that are placed on HDFS. The NameNode retains metadata regarding the location of each block within the cluster. Each block within the cluster is distributed and replicated across multiple data nodes to ensure the redundancy and efficiency of retrieving data. HDFS ensures that any data node failure will not result in data loss. For example, when HDFS receives a query, it first acquires metadata information about the location of the file from the NameNode, and then retrieves actual data blocks from the data nodes.

Hive

Originally developed by Facebook™ for data warehousing, Hive is now an open-source Apache project that uses an SQL-like language called HiveQL that generates MapReduce jobs running on the Hadoop cluster.

Hive works by turning HiveQL queries into MapReduce jobs that are submitted to the cluster. Hive queries functions on HDFS directories (or tables) containing one or more files, similar to an RDBMS. Hive also supports many formats for data storage and retrieval.

Hive can identify the structure and location of the tables, since they are specified when the table metadata is created. The metadata is then stored in the metastore within Hive and contained in an RDBMS such as Postgres or MySQL. The metastore contains the column format and location, which is exported by Hive. The query itself operates on files stored in a directory on HDFS.

MapReduce

MapReduce is a fault-tolerant method for distributing a task across multiple nodes where each node processes data stored on that node. Data is distributed automatically and parallel to each other.

The distribution and assigning of data to the nodes consists of the following phases:

- **Map.** Each Map task maps data on a single HDFS block and runs on the data node where the block is stored.
- **Shuffle and Sort.** This phase sorts and consolidates intermediate data from all Maps tasks and occurs after all Map tasks are complete but before the Reduce task occurs.
- **Reduce.** The Reduce task takes all shuffled and sorted intermediate data (for example, the output of the map task), and produces the final output.

Parquet	A Column-wise data serialization file format used in Hadoop. Parquet files can be queried with Hive. Use Parquet if you intend to only consume a few columns in a row.
Sqoop	Sqoop is a command-line interface application for transferring data between relational databases and Hadoop. It supports incremental loads of a single table or a free form SQL query as well as saved jobs which can be run multiple times to import updates made to a database since the last import. Imports can also be used to populate tables in Hive or HBase. The Sqoop (SQL to Hadoop) process transfers data between RDBMS and HDFS. For example, you can use Sqoop to import databases from Postgres, MySQL, and Microsoft SQL Server, then use Hadoop (which is in charge of transformation, aggregation, and long-term storage) to output BI batch reports. Sqoop performs the process very efficiently through a Map-only MapReduce job. It also supports JDBC, ODBC, and several specific direct database options such as PostGreSQL and MySQL.
YARN	Apache Hadoop YARN (Yet Another Resource Negotiator) is a cluster management technology. YARN is one of the key features in the second-generation Hadoop 2 version of the Apache Software Foundation's open source distributed processing framework. Originally described by Apache as a redesigned resource manager, YARN is now characterized as a large-scale, distributed operating system for big data applications. The YARN (MRv2) daemons consist of the following: ❏ **ResourceManager.** One per cluster, it starts ApplicationMasters and allocates resources on slave nodes. ❏ **ApplicationMaster.** One per job, it requests resources, manages individual Map and Reduce Tasks. ❏ **NodeManager.** One per slave (data node), it manages resources on individual slave nodes.

- **JobHistory.** One per cluster, it archives metrics and metadata about jobs that finished running (regardless of success).

Zookeeper

Apache ZooKeeper is a software project of the Apache Software Foundation, providing an open source distributed configuration service, synchronization service, and naming registry for large distributed systems. ZooKeeper's architecture supports high availability through redundant services.

Feedback

Customer success is our top priority. Connect with us today!

Information Builders Technical Content Management team is comprised of many talented individuals who work together to design and deliver quality technical documentation products. Your feedback supports our ongoing efforts!

You can also preview new innovations to get an early look at new content products and services. Your participation helps us create great experiences for every customer.

To send us feedback or make a connection, contact Sarah Buccellato, Technical Editor, Technical Content Management at *Sarah_Buccellato@ibi.com.*

To request permission to repurpose copyrighted material, please contact Frances Gambino, Vice President, Technical Content Management at *Frances_Gambino@ibi.com.*

Information Builders

iWay

iWay Big Data Integrator User's Guide

Version 1.5.2

DN3502221.0717

Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898