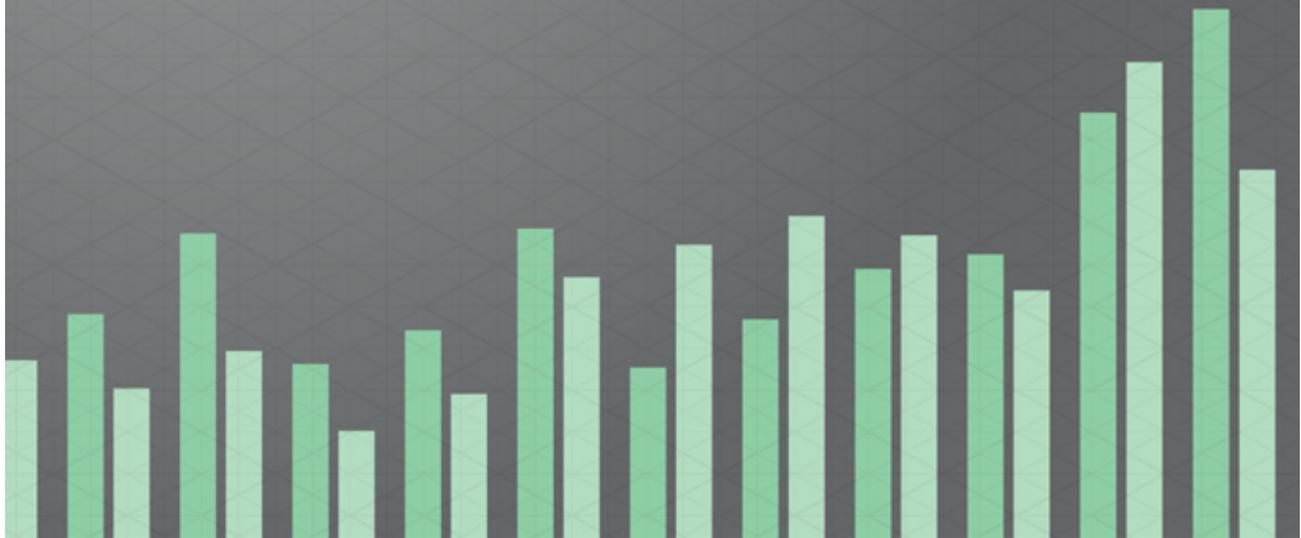




ACTUATE.  
The BIRT Company™



# BIRT Analytics



## Using BIRT Analytics Loader

Information in this document is subject to change without notice. Examples provided are fictitious. No part of this document may be reproduced or transmitted in any form, or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of Actuate Corporation.

© 2003 - 2015 by Actuate Corporation. All rights reserved. Printed in the United States of America.

Contains information proprietary to:  
Actuate Corporation, 951 Mariners Island Boulevard, San Mateo, CA 94404

[www.opentext.com](http://www.opentext.com)  
[www.actuate.com](http://www.actuate.com)

The software described in this manual is provided by Actuate Corporation under an Actuate License agreement. The software may be used only in accordance with the terms of the agreement. Actuate software products are protected by U.S. and International patents and patents pending. For a current list of patents, please see <http://www.actuate.com/patents>.

Actuate Corporation trademarks and registered trademarks include:  
Actuate, ActuateOne, the Actuate logo, Archived Data Analytics, BIRT, BIRT 360, BIRT Analytics, BIRT Data Analyzer, BIRT Performance Analytics, Collaborative Reporting Architecture, Dynamic Data Web, e.Analysis, e.Report, e.Reporting, e.Spreadsheet, Encyclopedia, Interactive Viewing, OnPerformance, Performancesoft, Performancesoft Track, Performancesoft Views, Quite4Me, Quiterian, Report Encyclopedia, Reportlet, The people behind BIRT, X2BIRT, and XML reports.

Actuate products may contain third-party products or technologies. Third-party trademarks or registered trademarks of their respective owners, companies, or organizations include:  
Mark Adler and Jean-loup Gailly ([www.zlib.net](http://www.zlib.net)): zlib. Apache Software Foundation ([www.apache.org](http://www.apache.org)): Axis2, log4, Tomcat. Boost.org: Boost libraries, licensed under the Boost Software License. CURL ([curl.haxx.se](http://curl.haxx.se)): Curl, licensed under a MIT/X derivate license. International Components for Unicode (ICU): ICU library. Marcin Kalicinski ([rapidxml.sourceforge.net](http://rapidxml.sourceforge.net)): RapidXML, licensed under the Boost Software License. Bruno Lowagie and Paulo Soares: iTextSharp, licensed under the Mozilla Public License (MPL). Math.NET: Math.NET, licensed under the MIT/X11 License. Microsoft Corporation: Access Database Engine, SQL Server Express. opencsv team ([sourceforg.net](http://sourceforg.net)): opencsv. openssl.org: OpenSSL, licensed under the OpenSSL license. qooxdoo.org: qooxdoo, licensed under the Eclipse Public License (EPL). Dave Scriven ([svg.codeplex.com](http://svg.codeplex.com)): SVG Rendering Engine, licensed under the Microsoft Public License. SQLAPI: SQLAPI++. [sqlite.org](http://sqlite.org): SQLite, public domain. [stlsoft.org](http://stlsoft.org): STLSoft libraries, licensed under the BSD license. Matthew Wilson and Garth Lancaster ([www.pantheios.org](http://www.pantheios.org)): Pantheios, licensed under a modified BSD license.

All other brand or product names are trademarks or registered trademarks of their respective owners, companies, or organizations.

Document No. 150731-2-580321 October 28, 2015

# Contents

<b>About <i>Using BIRT Analytics Loader</i></b> .....	<b>vi</b>
Accessing BIRT Analytics information .....	vii
Obtaining documentation .....	vii
Obtaining late-breaking information and documentation updates .....	vii
Obtaining technical support .....	viii
Supported and obsolete products .....	viii
<b>Chapter 1</b>	
<b>Introducing BIRT Analytics Loader</b> .....	<b>1</b>
About BIRT Analytics Loader .....	2
Understanding BIRT Analytics Loader .....	2
Understanding BIRT Analytics users .....	3
Getting started .....	3
<b>Chapter 2</b>	
<b>Working with projects</b> .....	<b>7</b>
About BIRT Analytics Loader projects .....	8
Planning a project .....	8
Understanding the FastDB repository .....	9
Creating a new project .....	9
Managing projects .....	11
Changing the feature tabs layout .....	12
Running a load process .....	13
Understanding data loading tasks .....	13
Running a project .....	13
Using reports .....	15
Setting database access privileges .....	15
Synchronizing data .....	16
Synchronizing data using the Post loading .....	16
Synchronizing data using BIRT Analytics Administration .....	17
Managing FastDB Engine accounts .....	18
<b>Chapter 3</b>	
<b>Loading data</b> .....	<b>21</b>
About loading data .....	22
Connecting to data .....	22
Connecting to databases .....	22
Updating data source definitions .....	23
Sharing data sources .....	23
Testing a Datasource connection .....	24
Using native drivers .....	25
Using ODBC drivers .....	27
Connecting to files .....	29
Loading local files .....	29
Loading remote files .....	30
Loading files using a file list .....	31
Creating a load process .....	32
About transactions .....	33
Setting load process options .....	33
Ignoring items during the load process .....	33

Creating a load transaction .....	34
Mapping data from files .....	34
Defining a target database .....	34
Defining a source file format .....	35
Defining table columns .....	37
Mapping data from databases .....	42
Defining a data set .....	42
Managing errors in the loading process .....	45
ON ERROR Attribute .....	45
<b>Chapter 4</b>	
<b>Managing data .....</b>	<b>47</b>
About managing data .....	48
Understanding Explorer .....	48
Viewing table properties .....	49
Exploring column properties .....	50
Understanding dynamic and permanent columns .....	52
Understanding Links .....	52
<b>Chapter 5</b>	
<b>Transforming data .....</b>	<b>55</b>
About transforming data .....	56
Understanding the Includes panel .....	56
Understanding scripts .....	57
Understanding the repository structure .....	58
Using the text editor .....	59
Creating new columns .....	59
AGGREGATE .....	60
DECODE .....	61
DOMAIN .....	63
EXPRESSION .....	66
Using LONGINT with DATE, TIME, or DATETIME arguments .....	71
Using INT with DATE, TIME, or DATETIME arguments .....	71
Using format patterns for DATE, TIME, or DATETIME values .....	71
Using regular expression patterns to match and replace text strings .....	73
INUMERIC .....	76
PARAMETRIC .....	78
QUANTILE .....	80
RANKING .....	81
UNPIVOT .....	83
Managing data objects .....	83
DROPAUTOLINKS .....	83
DROPCOLUMN .....	83
DROPDATABASE .....	84
DROPDERIVEDCOLUMNS .....	84
DROPLINK .....	84
DROPTABLE .....	85
INDEX .....	85
UNINDEX .....	85
LINK .....	85
RANGEDLINK .....	86
MAKEPERMANENT .....	86
RENAME .....	88
Miscellaneous instructions .....	88

EXPORT .....	88
IFEXISTS .....	94
IFLINKED .....	94
INCLUDE .....	96
ON ERROR .....	97
PRINT .....	97
VAR .....	97
<b>Chapter 6</b>	
<b>Administering BIRT Analytics Loader .....</b>	<b>101</b>
About BIRT Analytics Loader administration .....	102
Managing users .....	102
Scheduling load processes .....	103
Stopping the Tomcat service .....	105
Using External call .....	105
Running projects using qLauncher .....	106
Organization of storage in Loader .....	108
Copying a load project to a new location .....	109
Managing the repository .....	109
Creating a new repository .....	110
Moving a repository .....	111
Restoring a repository .....	112
Moving data objects to a different repository .....	112
<b>Glossary .....</b>	<b>113</b>
<b>Index .....</b>	<b>127</b>



# About Using BIRT Analytics Loader

---

BIRT Analytics provides fast, free-form visual data mining and predictive analytics. BIRT Analytics combines easy-to-use data discovery and data mining tools with powerful and sophisticated analytic tools.

BIRT Analytics consists of three key components:

- Actuate BIRT Analytics user interface, a web application that is used to carry out dynamic analyses
- BIRT Analytics Administration, a set of tools that supports administering user access and privileges
- BIRT Analytics Loader, a tool that extracts, transforms, and loads records from an external data source to FastDB, the BIRT Analytics data repository

*Using BIRT Analytics Loader* describes how to use BIRT Analytics Loader to collect data from various sources, transform the data depending on business needs, and load the data into a destination database.

This document is a guide for ETL (Extract, Transform, and Load) developers who understand basic concepts in working with data and are familiar with the data structure. Technical concepts and explanations about how to accomplish common activities are included in the following chapters. For more detailed information about any of the subjects discussed, see the complete set of Actuate documentation included with Actuate software or contact your database administrator.

*Using BIRT Analytics Loader* includes the following chapters:

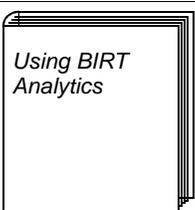
- *About Using BIRT Analytics Loader.* This chapter provides an overview of this guide.
- *Chapter 1. Introducing BIRT Analytics Loader.* This chapter describes how to access and log in to BIRT Analytics Loader.
- *Chapter 2. Working with projects.* This chapter describes how to create and run projects to load data.
- *Chapter 3. Loading data.* This chapter describes how to create data connections, data mappings and loading data into FastDB.
- *Chapter 4. Managing data.* This chapter explores the data elements, such as databases, tables, and columns, and their properties.
- *Chapter 5. Transforming data.* This chapter describes how to transform data for analysis.
- *Chapter 6. Administering BIRT Analytics Loader.* This chapter describes how to manage users and schedule data load processes.
- *Glossary.* This chapter provides definitions of terms used in the BIRT Analytics product and documentation.

---

## Accessing BIRT Analytics information

The online documentation includes the materials described in Table i-1. You can obtain HTML and PDF files from the Actuate website. These documentation files are updated in response to customer requirements.

**Table i-1** BIRT Analytics documentation

For information about this topic	See the following resource
Installing BIRT Analytics on Windows, Linux, and Mac OS X	 <p><i>Installing BIRT Analytics</i></p>
Overview of data analysis and data mining Using BIRT Analytics tools Visualizing data	 <p><i>Using BIRT Analytics</i></p>
Using BIRT Analytics Loader to extract, transform, and load data Using projects to manage data Administering BIRT Analytics Loader processes	 <p><i>Using BIRT Analytics Loader</i></p>
Using BIRT Analytics Admin to: <ul style="list-style-type: none"><li>■ Set up users and groups</li><li>■ Configure security</li><li>■ Configure and monitor system options</li></ul>	 <p><i>Administering BIRT Analytics</i></p>

## Obtaining documentation

Actuate provides technical documentation in PDF and HTML formats. You can download PDF or view HTML versions of the documentation from the following URL:

<http://developer.actuate.com/resources/documentation/birt-analytics>.

## Obtaining late-breaking information and documentation updates

The release notes contain late-breaking news about Actuate products and features. The release notes are available on the Actuate Support site at the following URL:

<http://support.actuate.com/documentation/releasenotes>

If you are a new user, you must first register on the site and log in to view the release notes. [actuate.com](http://actuate.com) also provides product update information.

---

## Obtaining technical support

You can contact Customer Support by e-mail or telephone. For contact information, go to the following URL:

<http://www.actuate.com/services/support/contact-support.asp>

## Supported and obsolete products

The Actuate Support Lifecycle Policy and Supported Products Matrix are available at the following URL:

<http://developer.actuate.com/resources/supported-products/birt-analytics/>



# 1

## Introducing BIRT Analytics Loader

This chapter contains:

- About BIRT Analytics Loader
- Understanding BIRT Analytics Loader
- Understanding BIRT Analytics users
- Getting started

## About BIRT Analytics Loader

BIRT Analytics Loader builds databases for data mining and analysis from data sources. The BIRT Analytics Loader tools enables you to maintain data quality by integrating and reconciling data across different systems, then deciding what data subsets to make available through the BIRT Analytics user interface.

BIRT Analytics Loader provides one, integrated tool for extracting, transforming, and loading (ETL) big data into BIRT Analytics. The ETL process pulls data from one, or multiple data sources and places it into another database, using the following processes:

- Extracting reads data from a database.
- Transforming converts extracted data from its original form into an analytical format stored in FastDB. Use features such as transform rules and lookup tables to organize data.
- Loading writes the data into the target FastDB database.

## Understanding BIRT Analytics Loader

BIRT Analytics Loader is a web-based application to create, execute, and schedule data loading processes. Figure 1-1 shows the BIRT Analytics architecture.

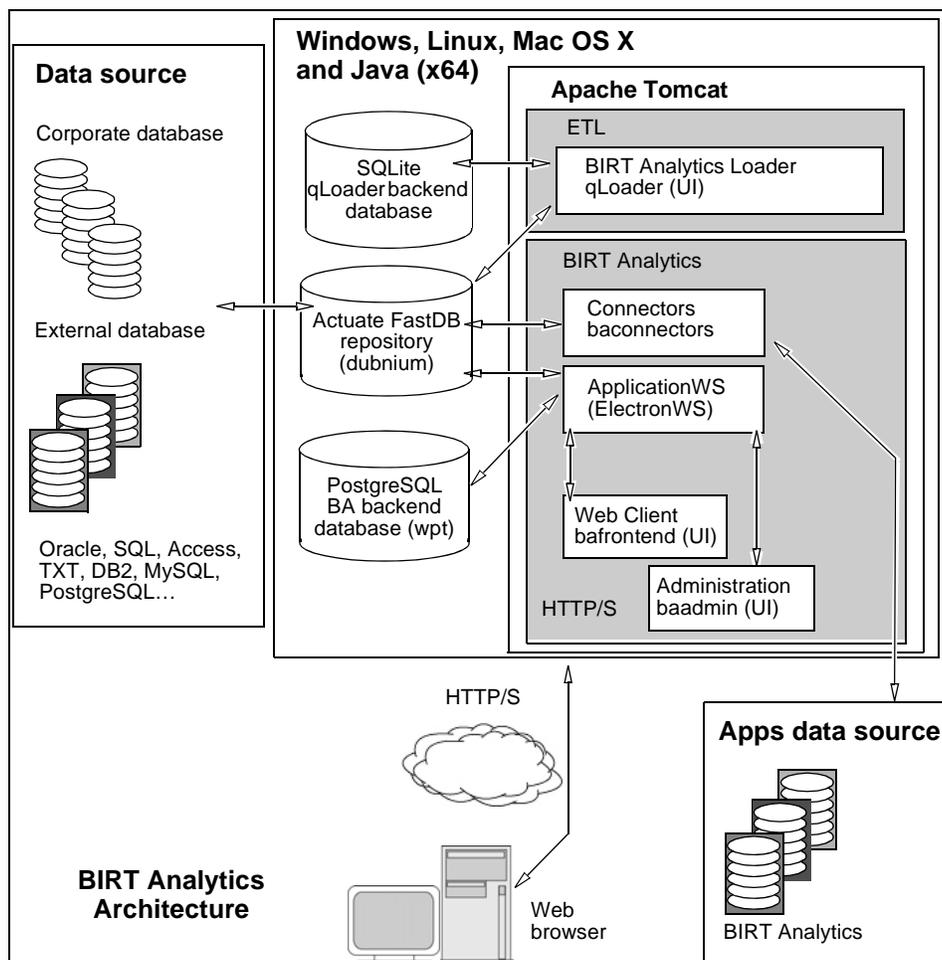


Figure 1-1 BIRT Analytics architecture

BIRT Analytics Loader supports importing data from multiple data sources, relational databases, and text files, using database native drivers and ODBC connectors. The application creates processes that extract, transform, and load data, storing it in a proprietary format in a FastDB repository. The FastDB repository contains different types of objects, such as databases, tables and columns. This repository is managed by the FastDB engine. The engine runs as a service, named BIRT Analytics - FastDB, also known as dubnium.

---

## Understanding BIRT Analytics users

Four types of users access Actuate BIRT Analytics products:

- **BIRT Analytics users**  
Users who log in to BIRT Analytics user interface to do data mining and analysis.
- **BIRT Analytics Administration users**  
These users manage access to objects, specify security filters, define password policies, browse logs, synchronize the database, specify password policies, and perform other security maintenance actions.
- **BIRT Analytics Loader users**  
Users who log in to BIRT Analytics Loader to create and execute projects for loading external data into the FastDB repository.
- **FastDB Engine users**  
The FastDB Engine user is the internal account the FastDB engine uses to communicate between the FastDB database and the rest of the internal modules. This account provides an additional layer of security for protecting sensitive data. When you create a project for loading data you must provide the secure account used by the engine to access the data. For more information on administering the FastDB engine data security account, see “Managing the repository” in Chapter 6, “Administering BIRT Analytics Loader.”

---

## Getting started

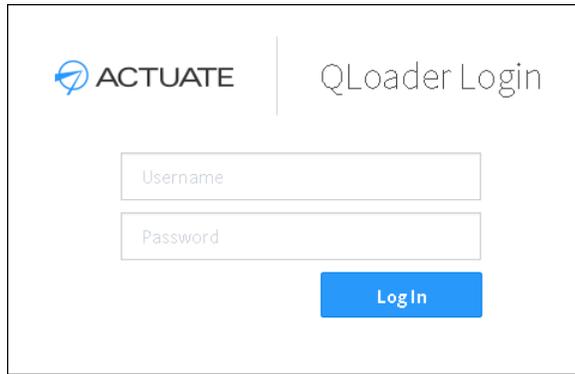
Open BIRT Analytics Loader to create projects that build database loading processes. Actuate BIRT Analytics Loader installs as part of the Actuate BIRT Analytics product. For more information, see the BIRT Analytics installation guide.

To access BIRT Analytics Loader, you need a web browser.

### How to log in to BIRT Analytics Loader

- 1 You open the BIRT Analytics Loader web application using a hyperlink or a web address similar to the following URL: `http://webserver:8110/qloader`.
  - `<webserver>:8110` is the web server and TCP port that runs the BIRT Analytics Loader application. Alternatively, you can use an IP address instead of the web server name.
  - `qloader` is the context root to the BIRT Analytics Loader application. The context root is set by the web server administrator when installing the application.

Figure 1-2 shows the login page for BIRT Analytics Loader.



**Figure 1-2** Logging in to BIRT Analytics Loader

2 To log in, provide the following login credentials:

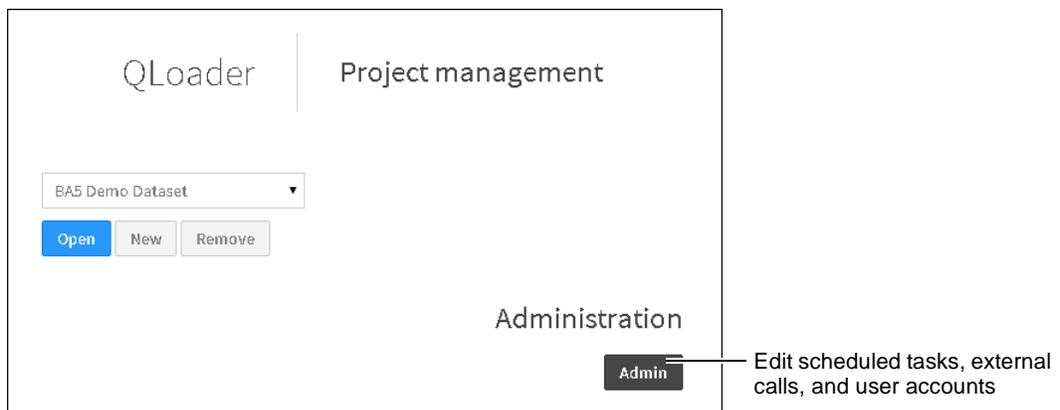
- In Username, type the user name.
- In Password, type the password associated with the user name.

During installation, BIRT Analytics Loader provides an administrator user account for the following web applications:

- BIRT Analytics user interface, also known as BAfrontend, account name Administrator
- Analytics Admin, also known as BAadmin, account name Administrator
- Loader, also known as Loader, account name Administrator
- FastDB engine, also known as Dubnium, account name SYSTEM

The password for these administrator accounts is PASSWORD. Use the Loader Admin to modify the loader administrative account name and password.

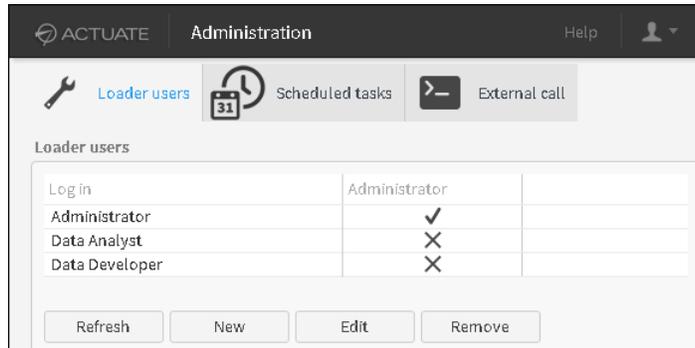
3 Choose Log In. The BIRT Analytics Loader appears, as shown in Figure 1-3.



**Figure 1-3** Viewing the BIRT Analytics Loader home page

After logging in to BIRT Analytics Loader, you can start managing projects. You can choose a project to open, remove a project, or create a new project. Each project is a collection of all objects added to the FastDB repository, such as data from databases, files, and web services.

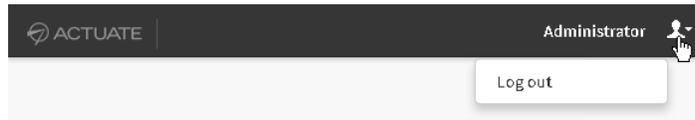
An administrator can use Admin to manage BIRT Analytics Loader user privileges and schedule loading processes, as shown in Figure 1-4.



**Figure 1-4** Viewing the Loader Admin options

### How to log out of Actuate BIRT Analytics Loader

Choose Log out to close a user session of BIRT Analytics Loader, as shown in Figure 1-5.



**Figure 1-5** Logging out of BIRT Analytics Loader



# Working with projects

This chapter contains:

- About BIRT Analytics Loader projects
- Planning a project
- Understanding the FastDB repository
- Creating a new project
- Managing projects
- Running a load process
- Synchronizing data
- Managing FastDB Engine accounts

---

## About BIRT Analytics Loader projects

BIRT Analytics Loader organizes data loading activities into projects. Projects are logical descriptions of how to extract, load, and transform the data into the target database. A BIRT Analytics Loader User (non-administrator) works with only the projects they create. The BIRT Analytics Loader Administrator can access projects that other users have previously shared with the administrator.

A project that completes a typical Extract, Transform, Load (ETL) process includes completing each of the following activities, as illustrated in Figure 2-1:

- Defining data sources
- Mapping the data
- Defining the data transformation
- Scheduling the load processes to run

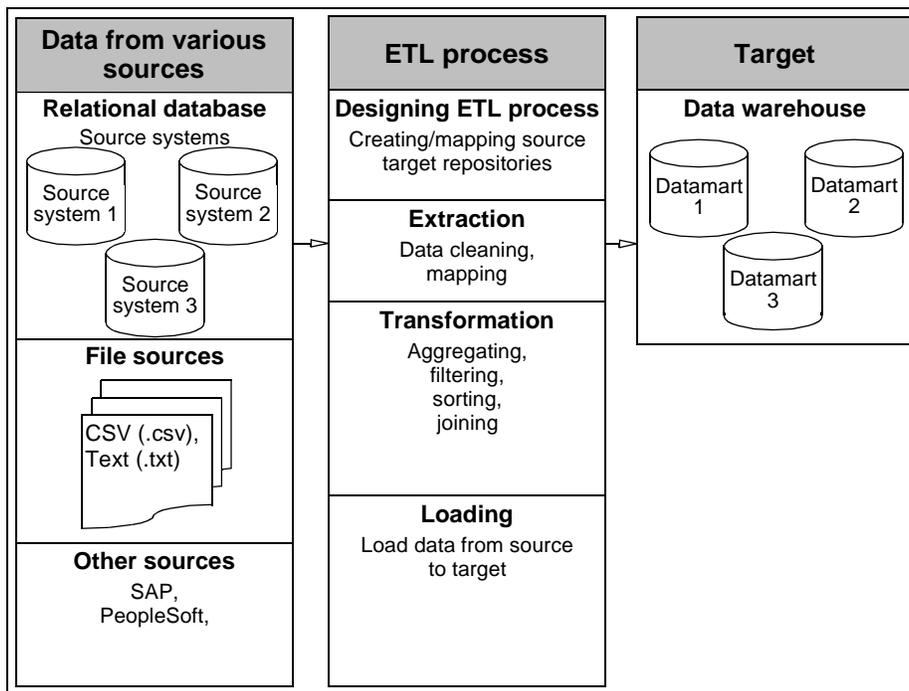


Figure 2-1 Sample ETL process

---

## Planning a project

Typically, there are three key players involved in building an ETL process: a database administrator, a data modeler, and an ETL developer. Sometimes, one person fills multiple roles. The key factor for success when creating an ETL process is to diligently plan the project before starting the ETL process development.

Planning a project requires:

- Knowing your data sources and completing a detailed mapping specification. A successful ETL process depends entirely on a completed mapping specification.
- Documenting how you are going to extract, clean, and transform your data sources to add to your database table definitions.

- Establishing rules that define how to update database tables.

This is an important factor in the ETL design. Depending on the nature of the database, the update is as small as a data refresh, cleaning old data, and loading new data. In some occasions, the update requires appending data based on complex condition such as keyed reads or joins against an existing dimension before writing the new or updated rows to a database table. Building these rules can potentially add days or weeks to your project timeline.

In the planning process you must define:

- Input and output record layouts
- The location of source and target files and databases
- File and table sizing information
- File and table volume information
- Documentation on how to transform the data

---

## Understanding the FastDB repository

BIRT Analytics Loader loads data into a FastDB database repository, operated by the FastDB Engine. When FastDB is installed, a data repository is created in this path:

```
C:\Program files\BIRTAnalytics\data\FastDB\databases
```

You can use the same repository to create and store your databases. BIRT Analytics enables you to create a different location for your database repository and set up BIRT Analytics Loader to work with this location. For more information on how to create and set up a new data repository, see “Managing the repository” in Chapter 6, “Administering BIRT Analytics Loader.”

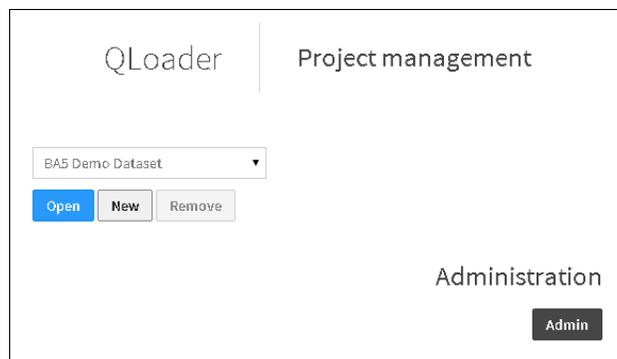
---

## Creating a new project

Use the BIRT Analytics Loader to complete the following tasks:

- Create a new project
- Open an existing project
- Remove an existing project

Figure 2-2 shows a demo data project.



**Figure 2-2** Creating a new project

Choose New to create a new project. This opens Create new project, as shown in Figure 2-3.

The screenshot shows a dialog box titled 'QLoader' with a sub-header 'Create new project'. It contains the following fields and controls:

- Name:** An empty text input field.
- IP:** An empty text input field.
- Port:** An empty text input field.
- SSL/TLS:** A checkbox that is currently unchecked.
- User:** A text input field containing the text 'SYSTEM'.
- Password:** A text input field with asterisks (\*\*\*\*\*).
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

**Figure 2-3** Creating a new project

BIRT Analytics Loader users can access only projects they create. A BIRT Analytics Loader administrator can also only access his own projects, except in the case of other users that have previously shared a project with him.

**How to create a new project**

- 1 In Name, type the name of the project. In our example it is called *MyFirstProject*. Figure 2-4 shows the new project options.

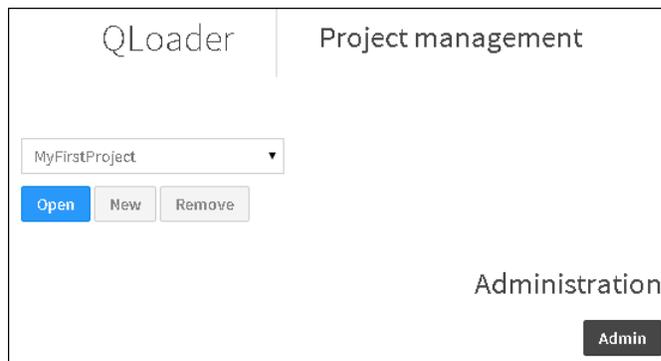
This screenshot shows the same dialog box as Figure 2-3, but with the following values entered:

- Name:** MyFirstProject
- IP:** localhost
- Port:** 8105
- SSL/TLS:** Unchecked
- User:** SYSTEM
- Password:** \*\*\*\*\*

**Figure 2-4** Setting up MyFirstProject

- 2 In IP, type the IP address or name of the server where the FastDB engine is installed.
- 3 In Port, type the port number that the FastDB engine uses. The default port number is 8105.
- 4 If you choose SSL/TLS, a secure connection is used. The default port number is 8106.
- 5 In User, type the FastDB Engine user. The default user is SYSTEM.
- 6 In Password, type the FastDB Engine User password. The default password is PASSWORD.

- Choose OK to create the project. Project management page appears. Now choose your project in the drop-down list and choose Open, as shown in Figure 2-5.

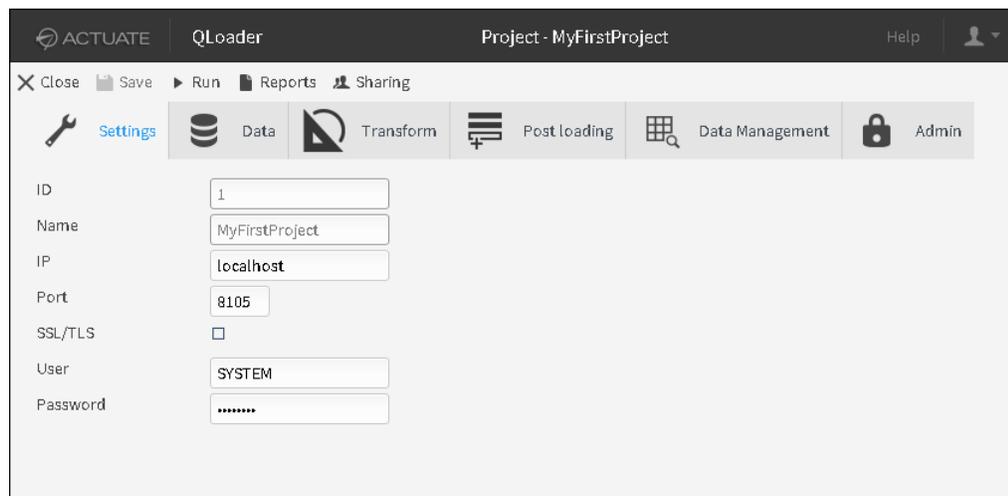


**Figure 2-5** Opening an existing project

## Managing projects

After opening a project using BIRT Analytics Loader, the *project development page* appears. This page contains a top Banner, a Control menu, Feature tabs, and a Work area. The top banner shows the *project name* and the *Help* and *Logout* links.

This page is shown in Figure 2-6. The Control menu options are described in Table 2-1 and the Features tabs in Table 2-2. The Work area is the large open space on the right.



**Figure 2-6** Developing a project

Table 2-1 describes the control menu item links available to you for managing a project.

**Table 2-1** Control menu icons

Icon	Menu items	Purpose
	Close	Closes the current project, after confirmation, and returns to the Projects management window.
	Save	Saves the project. To make any change effective, it is necessary to save data by choosing Save.
	Run	Executes the load process.

**Table 2-1** Control menu icons

Icon	Menu items	Purpose
	Reports	Shows the load process log files.
	Sharing	Displays the list of Loader users that you have created, showing whom you can share your project with.

The feature tabs contain links to the main project activities described in Table 2-2.

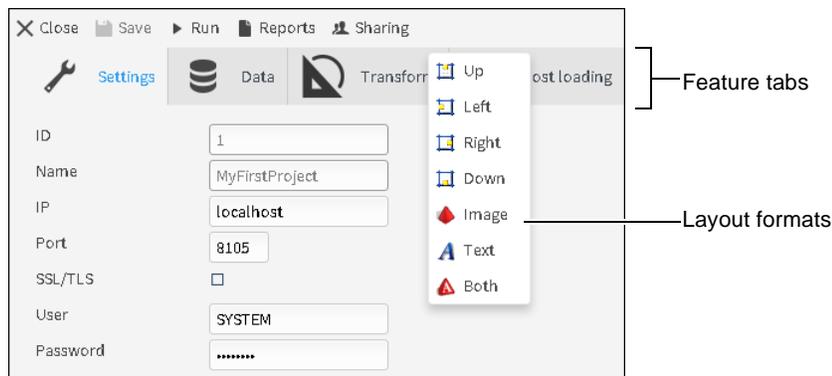
**Table 2-2** Icons for the feature tabs

Icon	Label	Purpose
	Settings	The tool used to change the configuration parameters of the current project. These parameters are the same as those set in Create new project. You cannot change the project name.
	Data	The tool to set up the connections to the data sources and define the mappings to the tables in the target database.
	Transform	Tools for defining the data transformation rules.
	Post loading	Tool for automating a synchronization process after loading data into the FastDB and clearing the cache. Requires the URL of the ApplicationWS, also known as electronws, server and the required actions.
	Data Management	Tools used to browse and explore data in the target database, and create and remove links between data columns in tables.
	Admin	Supports administering the FastDB Engine users. An administrator can create a new engine user, change a password, and delete it. You must provide the following information: login name, password, password confirmation, and old password.

For security reasons, request every user to log out before closing the browser.

## Changing the feature tabs layout

You can change the presentation of feature tabs using the feature tab menu, as shown in Figure 2-7. These options support placing the tabs in different positions and displaying text, an icon, or both on each tab. To access these options, right-click any of the feature tabs.



**Figure 2-7** Setting display options for feature tabs

---

## Running a load process

You start a load process by using a project in BIRT Analytics Loader. Before running a project, you must create the data source connections and define the loading transactions and the data transformation scripts. To make the loaded data available to the BIRT Analytics users, you must synchronize the loaded database with the active database and grant access privileges to database users.

After defining the table mappings and all transactions in a load process, you can run the project to load the data. You can run a single or multiple transaction processes. You can run a process with or without data transformation.

The load process loads the data in the FastDB database, the same database that users analyze. To avoid inaccurate analysis, consider loading data when users are not logged in to the system.

To prevent user access to databases while a load process runs, stop the web server running the BIRT Analytics application. Alternatively, schedule the loading process to run at a time when users are off line. For more information on how to schedule a load process, see “Scheduling load processes” in Chapter 6, “Administering BIRT Analytics Loader.”

## Understanding data loading tasks

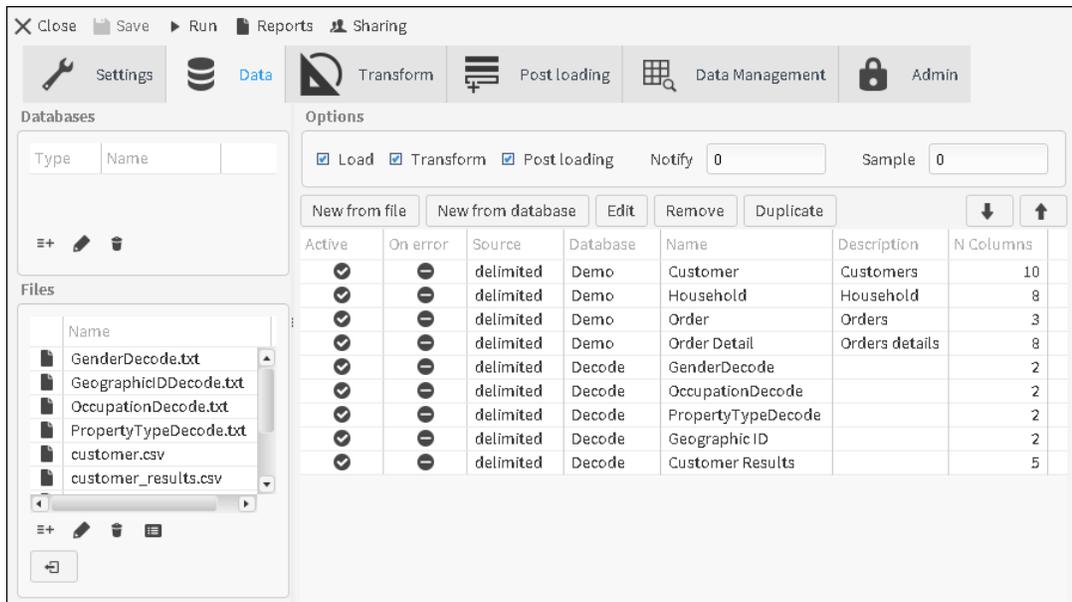
The following tasks are part of a typical load process to load data into the FastDB engine:

- Define a data loading project to collect, organize, and load data into FastDB.
- Run a data loading project.
- Remove cached data objects from FastDB. You can run this as a Loader—Post loading action or in BIRT Analytics Admin.
- Synchronize the FastDB repository with the ApplicationWS backend database (also known as wpt). You can run this as a Loader—Post loading action or in BIRT Analytics Admin.
- Set access permissions for repository objects if structural changes have occurred in the database, (such as changing, adding or removing database objects). Use BIRT Analytics Admin to set permissions. See “Setting database access privileges”, later in this chapter, for more information about setting object permissions.

Skipping one of these steps can cause users of the BIRT Analytics user interface to see an empty Data tree.

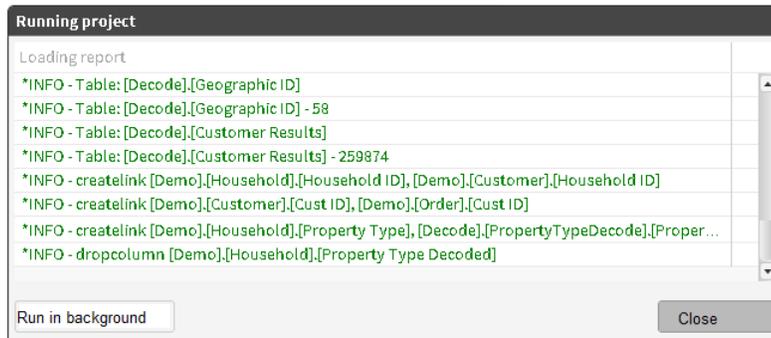
## Running a project

To run a loading project, open the Data tab and make any desired changes. Then choose Run, as shown in Figure 2-8. Use the other feature tabs to add data transformations and post loading actions to the loading project.



**Figure 2-8** Reviewing the Data feature

In the message that confirms whether you want to run the project, choose Yes. A list of database table names and the records loaded into each table appears in Running project, as shown in Figure 2-9.



**Figure 2-9** Running project log

The Running project log provides information about the tables into which the data is loaded, and the number of the records that are loaded. In the log viewer, standard message text is in black print, INFO messages in green and Error messages in red. The stack trace shows the line of script where the error was generated.

To limit the number of message lines in the log, use Options—Notify. Set notify to zero to record a log message at the end of a load process for each table after all rows for the table finish loading. Set the notify value to record a log message every time the number of loaded rows equals the notify value. For example, with notify set to 100000, a log message is generated every time 100000 rows are loaded. In this case, a data resource with 1000000 rows of data generates 10 messages in the log.

For loading processes requiring a long time, choose the option Run in background. This option minimizes the Running project window and enables you to work on a different project until the running project finishes.

## Using reports

To browse through the generated log files, select Reports from the control menu. The list of reports appears in Executions, as shown in Figure 2-10. Date shows a date and time when a project is executed. State shows the generation status. After selecting a date, the log content appears in Detail.

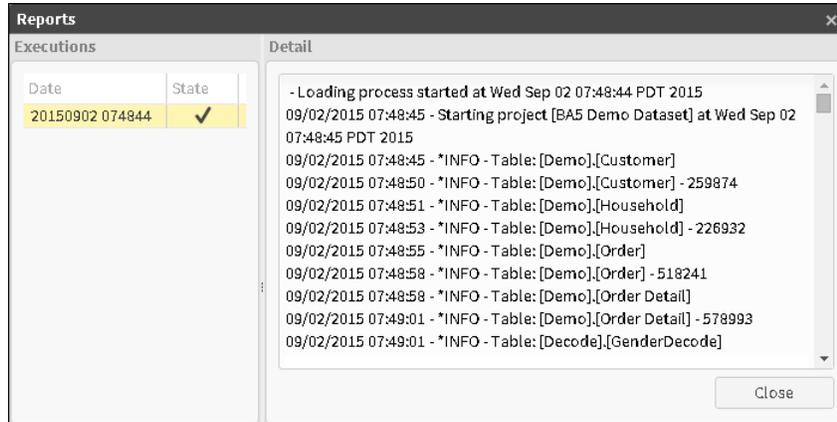


Figure 2-10 Browsing loading execution reports

## Setting database access privileges

When a load process causes structural changes to occur in the database, such as changing, adding or removing database objects, set access privileges for the users or groups using the BIRT Analytics Admin tool. Figure 2-11 shows user groups.

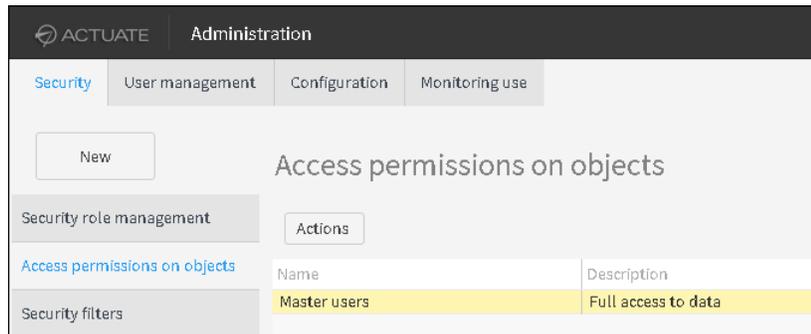
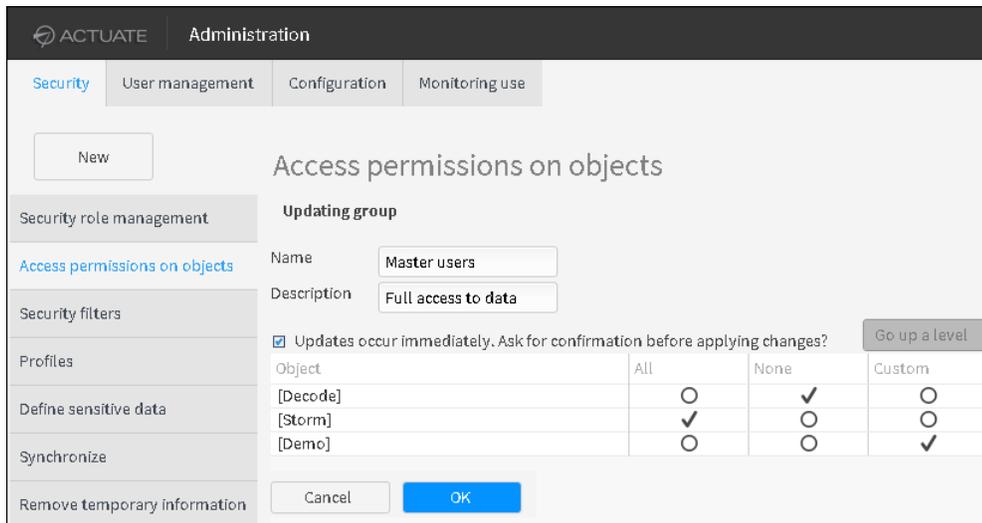


Figure 2-11 Setting data access permissions in Administration

Select a user group and choose Actions - Modify.

Figure 2-11 shows example data objects and access permissions. You can set permissions for entire databases and for specific data objects within a database.



**Figure 2-12** Setting data access permissions in Administration

For more information about setting repository data object permissions, see the administration guide *Administering BIRT Analytics*.

## Synchronizing data

Update the application database to contain the most recent changes in the FastDB Engine repository by synchronizing the repository with the data. The following tasks are done after loading data:

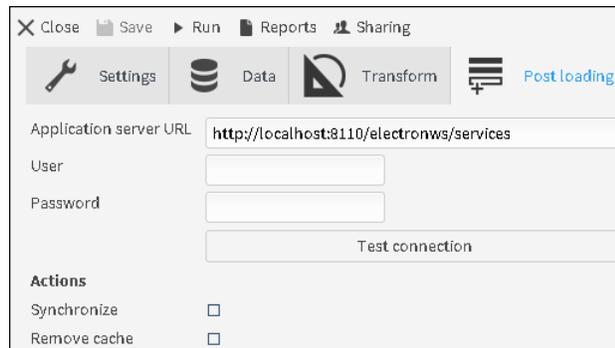
- Synchronize data to synchronize the FastDB and ApplicationWS database after loading data
- Remove cache to remove the old data structure

Synchronization is required after making a change in the analytical repository that alters the database structure, such as the addition, modification, or deletion of a column, table, or other data object. In the same way, it is also required when you change the data, after increasing, reducing, or modifying the records in a database.

To synchronize data, and remove temporary information after a load, use the Post Loading process in BIRT Analytics Loader. You can also use the BIRT Analytics Admin tool.

### Synchronizing data using the Post loading

BIRT Analytics Loader provides a Post loading option located in the *Post loading tab*, shown in Figure 2-13. Use Post loading to schedule the synchronization process to run when data loading finishes. You can also clear the cache after data load to temporary information.



**Figure 2-13** Post loading tab

Post loading requires the URL of the ApplicationWS server, authentication and the requested actions. After your data is loaded in to FastDB, use the BIRT Analytics Administration tool to assign access permissions on data objects.

Enable Post loading—Actions—Synchronize to synchronize the FastDB and ApplicationWS database after loading data.

Enable Post loading—Actions—Remove cache to remove the old data structure.

## Synchronizing data using BIRT Analytics Administration

You can use BIRT Analytics Administration to complete the tasks done after loading data, such as synchronizing data objects, removing temporary information and setting data object access permissions. The BIRT Analytics Administration is available at the following URL:

`http://localhost:8110/baadmin`

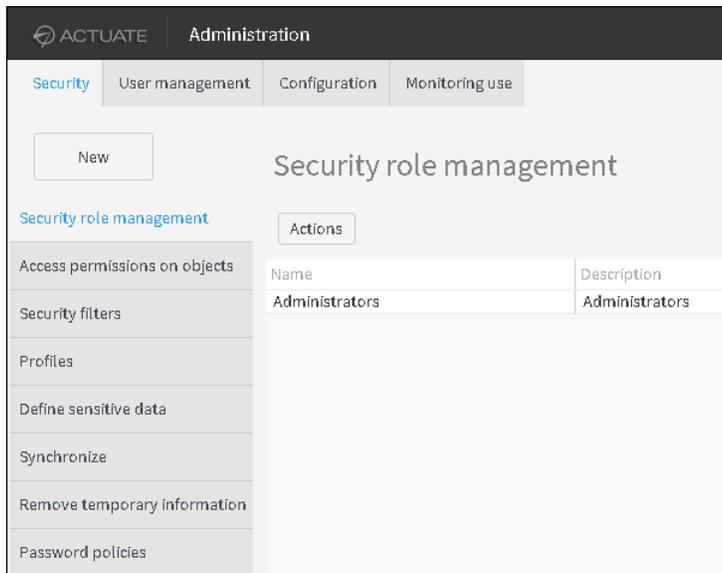
Replace localhost with the name or IP address of the computer where you installed BIRT Analytics when you access BIRT Analytics from another computer on your network.

The *Synchronize* option enables the administrator to update the application database to add the most recent changes in the BIRT Analytics Engine repository. You can also do this with the post loading feature of Loader.

The *Remove temporary information* option is used for clearing the cache after a loading process has finished. You can also do this with the post loading feature of Loader.

The *Access permissions on objects* option is used for setting database access privileges for users and user groups.

Figure 2-14 shows the *synchronize* option, along with the *Remove temporary information* and *Access permissions on objects* options available in the “Security” tab of BIRT Analytics Administration.

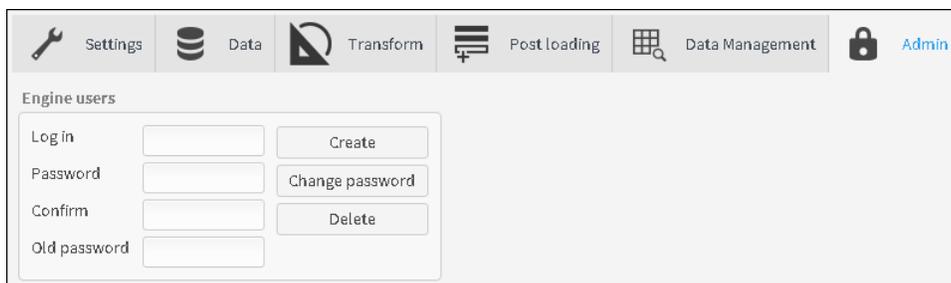


**Figure 2-14** Reviewing data object permissions in Administration

## Managing FastDB Engine accounts

Dubnium server uses an internal account to communicate with the FastDB database and the rest of the internal modules. This account provides a security layer that protects sensitive data. The administrator who runs a loading project must create the secure account and login credentials for the account. BIRT Analytics Loader passes a project’s user credentials to the FastDB Engine at run time. The engine authenticates the credentials against the existing secure database accounts. If an account does not exist or a password is not the same, a project fails to run and load data. This feature enables the system administrator to control user access to sensitive data by changing passwords or deleting FastDB Engine user accounts.

Click on the “Admin” tab in BIRT Analytics Loader to manage FastDB Engine users in the “Engine users” screen that appears, as shown in Figure 2-15.



**Figure 2-15** Managing FastDB Engine users

Use these Admin features to perform the following account creation tasks:

- **Create**  
Create a new user account.
- **Change password**  
Change a password of an existing user account.
- **Delete**  
Delete an existing user account.

To create and manage each FastDB Engine user, perform the following tasks in each field:

- Login  
Define a user name.
- Password  
Define a password.
- Confirm  
Confirm the password. You use this property when you create a new user or change a password for an existing user.
- Old password  
Define an old password. Required when you change a password.



# 3

## Loading data

This chapter contains:

- About loading data
- Connecting to data
- Creating a load process
- Creating a load transaction

---

## About loading data

The load process is composed of two main steps: loading and transformation. The configuration of the load process is done in the *Data* tab of the BA Loader main *Project* screen.

The Data page, accessed by clicking on the *Data* tab, is divided into three panels: Databases, Files and Options, as shown in Figure 3-1. The Databases and Files panels are where you define the connections to the data sources. The Options panel is where you define a load process. These options use transactions to define how a load process runs. A load transaction describes the data mappings from the data source to the target database. Each load process can contain multiple transactions.

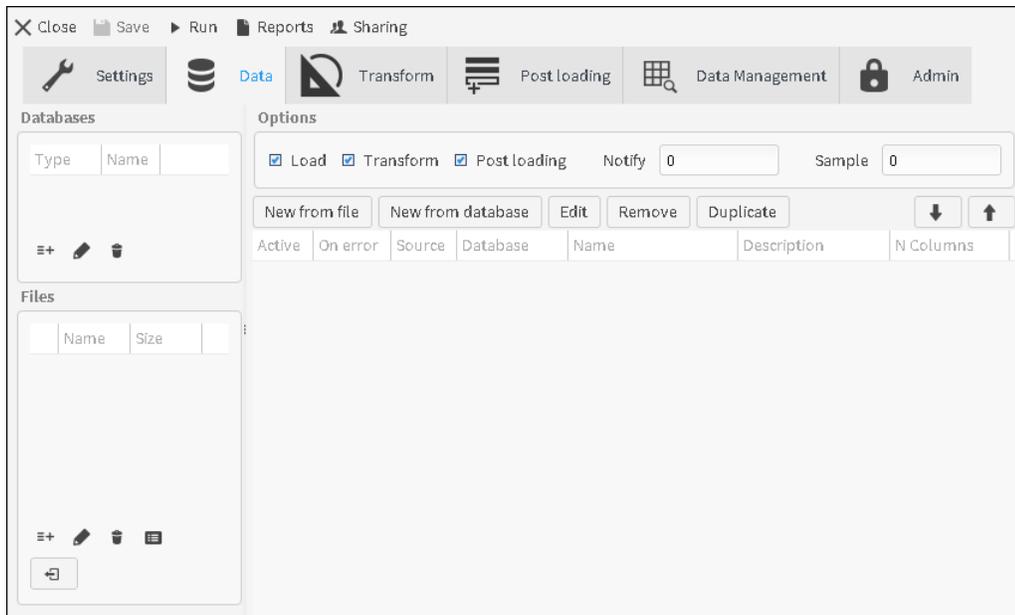


Figure 3-1 Navigating databases and load processes

---

## Connecting to data

BIRT Analytics Loader supports connections to relational databases and flat files. Data source connections are accessible across multiple projects associated with the same user. File connections are associated only with the project where they are defined. You cannot share file connections across multiple projects.

### Connecting to databases

You can create and name any number of data source connections for your load process. The data sources can be of different types. For example, a load process can use data from a database and data from a flat file repository. Each type of data source requires different connection information.

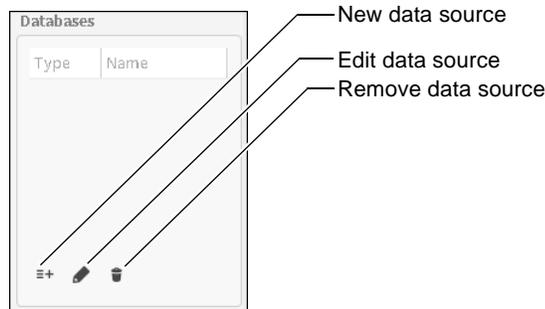
BIRT Analytics Loader connects to databases using both native and ODBC drivers. BIRT Analytics Loader supports native drivers for the following database types:

- Microsoft SQL Server
- MySQL Database Server
- Oracle Database

## ■ PostgreSQL

Use an ODBC driver to connect to a data source not supported by BIRT Analytics Loader. After installing the driver and configuring an ODBC connection, you can access data from the data source.

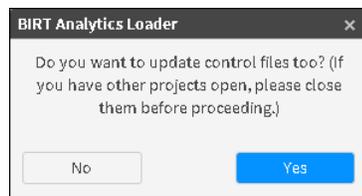
In BIRT Analytics Loader you can define a new data source, edit an existing one, or remove a data source by clicking on the corresponding icon, as shown in Figure 3-2.



**Figure 3-2** Managing data sources

## Updating data source definitions

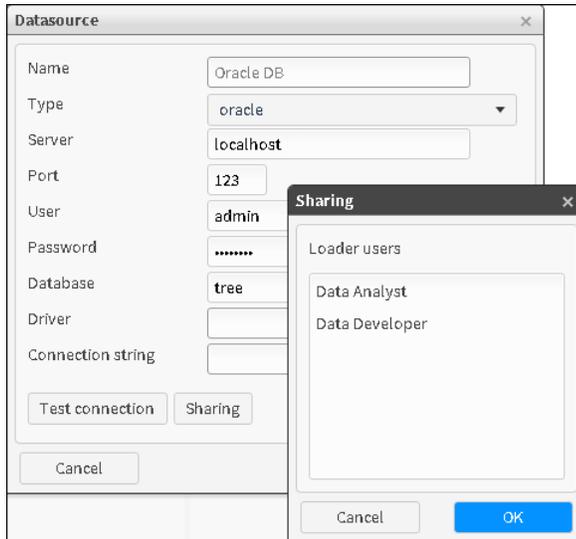
Choose *Edit data source* to update data source definitions. You also need to update the related table definitions. When you save your changes after updating, the following message appears asking you to confirm the updating operations, as shown in Figure 3-3.



**Figure 3-3** Updating data source definitions and their related tables

## Sharing data sources

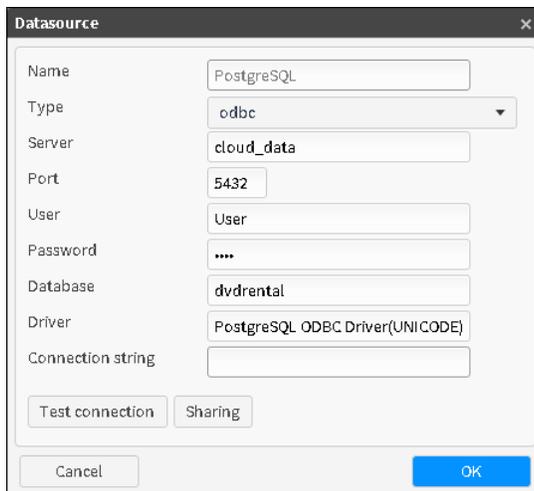
This option makes it possible to share data sources with other Loader users, whether using ODBC or Native drivers. This is done via the "Sharing" button which is available when you are editing an existing data source definition. Clicking on the "Sharing" button opens a window where you can select the Loader users that you want to share with. (See Figure 3-4).



**Figure 3-4** Sharing data source access with selected users

### Testing a Datasource connection

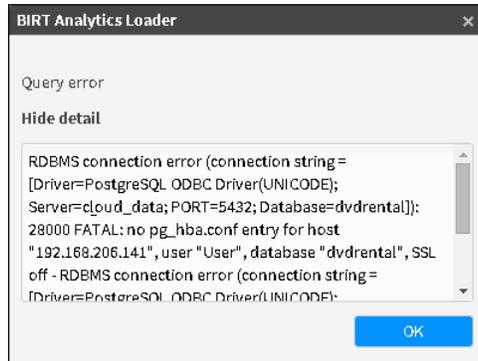
You can test the connection to an external database when you are creating or editing an existing data source. Figure 3-5 shows the option to test a database connection.



**Figure 3-5** Testing a data source connection

Errors returned from the database are displayed. Figure 3-6 shows the results of a connection error to a PostgreSQL database configuration. After correcting the error you can retry the connection test. Other error types can be:

- Authentication error
- Host not found
- Server not responding
- ODBC driver not found
- Native database client not found



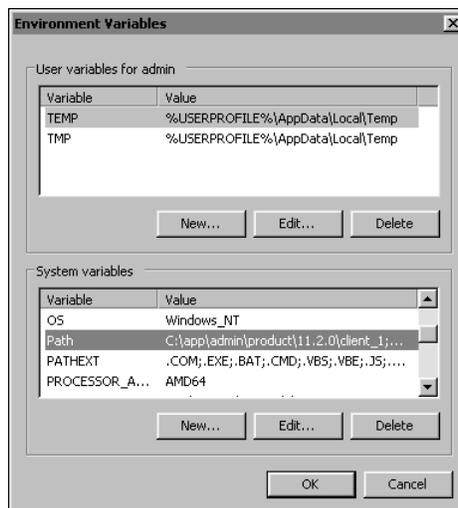
**Figure 3-6** Reviewing a database connection error

## Using native drivers

Before creating a connection using a native driver confirm the driver is installed on the computer running Loader. For example, verify if the path to the driver is in the System Path. To see the System Path using the Windows command line use the following command:

```
echo %PATH%
```

To see the System Path using the Windows Control Panel → System → Advanced System Settings → Environment Variables → Path, as shown in Figure 3-7.



**Figure 3-7** Editing Path variable

To create a data source connection, provide the database connection information, as shown in Table 3-1. Contact your database administrator to obtain this information.

**Table 3-1** Database connection properties

Property	Description
Name	The name of the database connection, which must be unique
Type	The type of connection, selected from the values in the drop-down list
Server	The database server, which can be a machine name or an IP address
Port	The connection port
User	A valid user name to connect to the database
Password	A valid password to connect to the database
Database	The name of the database

**Table 3-1** Database connection properties (continued)

Property	Description
Driver	The driver name, required for ODBC drivers only
Connection string	Custom connection properties, required for custom ODBC drivers only

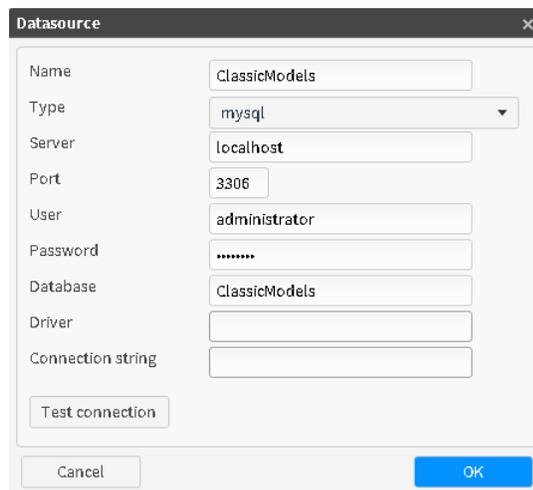
### Downloading native database drivers

The following locations contain native database drivers:

- MS SQL Server client software, `SQLServer2008R2SP2-KB2630458-x64-ENU.exe`  
`http://prod.actuate.com/birt-analytics-52/mssql`
- MySQL client software, `mysql-5.6.21-winx64.zip`  
`http://prod.actuate.com/birt-analytics-52/mysql`
- PostgreSQL client software, `postgresql-9.0.18-2-windows-x64.exe`  
`http://prod.actuate.com/birt-analytics-52/postgresql`
- Oracle client software, `ODAC1120320_x64.zip`  
`http://prod.actuate.com/birt-analytics-52/oracle`

### How to connect to a database using a native driver

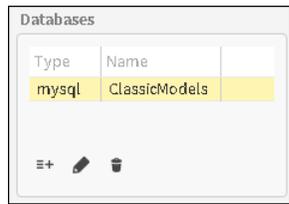
-  1 In Databases, choose the *New datasource* icon. Datasource appears, as shown in Figure 3-8.



**Figure 3-8** Using a native driver

- 2 In Datasource, define the following properties:
- In *Name*, type a unique name. Using a name that describes the database content helps identify different databases you define as data sources.
  - In *Type*, select the source database type from the list of available types. In this example, `mysql` is the source database type.
  - In *Server*, type the name of the server that hosts the source database. In this example, an IP address identifies that server.
  - In *Port*, type the connection port. The `mysql` database installation in this example uses the default port `3306`.
  - In *User*, type a valid user name defined for the source database.
  - In *Password*, type a valid password defined for the source database.
  - In *Database*, type the name of a database that contains source data.
  - In *Driver*, optionally type a native driver name, or type no characters. A connection that uses a native driver requires no name.
  - In *Connection string*, optionally add any parameters you want to send to the data source.

- 3 Choose *OK* to create the named data source connection. A new data source connection name and type appear in the *Databases* panel, as shown in Figure 3-9.

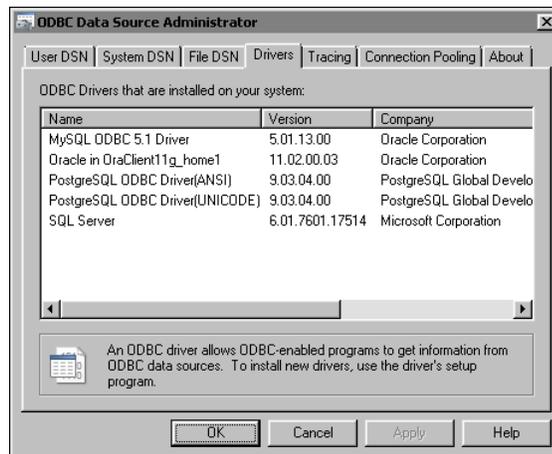


**Figure 3-9** Reviewing a data source from a native database driver

## Using ODBC drivers

Creating a connection using an ODBC driver is very similar to the native driver connections. The difference is that you must provide a driver name as a connection property. As well as the predefined ODBC data source types, a custom ODBC data source is available. To use the custom ODBC data source, you must type the required connection parameters as a text value.

To determine the name of your ODBC driver, open Control Panel → ODBC Data Source Administrator. The driver name appears in the Drivers tab, as shown in Figure 3-10.



**Figure 3-10** Finding an ODBC driver name

## Downloading ODBC drivers

The following locations contain ODBC database drivers:

- MS SQL Server client software, `msodbcsql.msi`  
<http://prod.actuate.com/birt-analytics-52/mssql-odbc>
- MySQL client software, `mysql-connector-odbc-5.1.13-winx64.msi`  
<http://prod.actuate.com/birt-analytics-52/mysql>
- PostgreSQL client software, `psqlodbc_09_00_0310-x64.zip`  
<http://prod.actuate.com/birt-analytics-52/postgresql>
- Oracle client software, `ODAC1120320_x64.zip`  
<http://prod.actuate.com/birt-analytics-52/oracle>

## How to connect to a database using an ODBC driver

- 1 In Databases, choose the + sign. Datasource appears.
- 2 Define the following properties for a standard ODBC data source, as shown in Figure 3-11:

The screenshot shows a 'Datasource' dialog box with the following fields:

- Name: SFData
- Type: odbc
- Server: CloudHosting
- Port: 1433
- User: administrator
- Password: .....
- Database: Demo
- Driver: SQL Server
- Connection string: (empty)

Buttons: Test connection, Cancel, OK

**Figure 3-11** Using an ODBC driver

- In *Name*, type a unique name. Using a name that describes the database content helps identify different databases you define as data sources.
- In *Type*, select odbc from the list of available connection types.
- In *Server*, type the name of the server that hosts the source database. In this example, a machine name, CloudHosting, identifies that server.
- In *Port*, type the connection port. The mssql database installation in this example uses the default port, 1433.
- In *User*, type a valid user name defined for the source database.
- In *Password*, type a valid password defined for the source database.
- In *Database*, type the name of a database that contains source data.
- In *Driver*, type the driver name, as defined in the ODBC Data Source Administrator. In this example, the name of the ODBC Microsoft SQL Server driver is SQL Server.

Alternatively, for a custom ODBC data source, define the following properties, as shown in Figure 3-12:

The screenshot shows a 'Datasource' dialog box with the following fields:

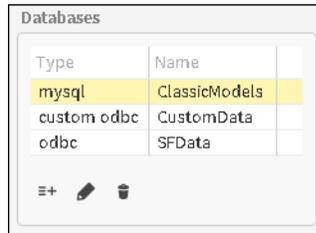
- Name: CustomData
- Type: custom odbc
- Server: (empty)
- Port: (empty)
- User: Administrator
- Password: .....
- Database: customdata
- Driver: (empty)
- Connection string: Driver={PostgreSQL UNICODE};Serv

Buttons: Test connection, Cancel, OK

**Figure 3-12** Custom ODBC driver

- In *Name*, type a unique name. Using a name that describes the database content helps identify different databases you define as data sources.
- In *Type*, select Custom ODBC from the list of available connection types.
- In *User*, type a valid user name defined for the source database.
- In *Password*, type a valid password defined for the source database.

- In *Database*, type the name of a database that contains source data.
  - In *Connection string*, type the parameters required to connect to the custom data source.
- 3 Choose *OK* to create the named data source connection. A new data source connection name and type appear in the *Databases* panel, as shown in Figure 3-13.



**Figure 3-13** Reviewing Database connections

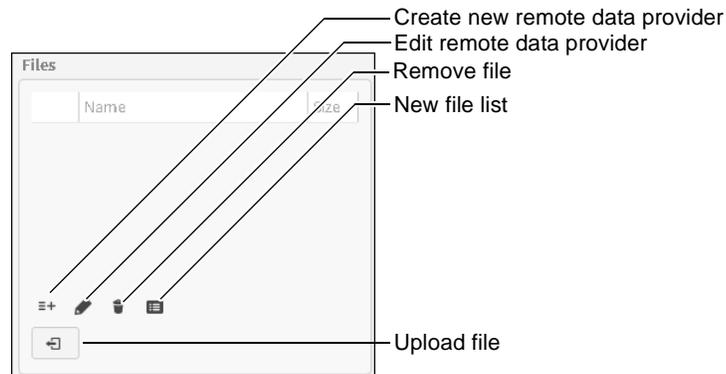
## Connecting to files

BIRT Analytics Loader can access data from a text file, or flat file, that conforms to a defined and consistent structure. These files are typically generated by business systems and applications that create logs. These files can also be spreadsheets saved as comma-separated values (CSV) files. Before using text file data in a load process, confirm the file uses a valid structure. There is no limit to the size of text files.

BIRT Analytics Loader supports the following options for defining flat file data sources:

- Loading a file from a local system folder
- Loading a CSV file from a remote system using FTP or HTTP
- Loading multiple files using a file list

In the *Files* panel located in the *Data* tab of your BIRT Analytics Loader project, you can manage data uploads from data files stored on local and remote systems, using the toolset shown in Figure 3-14.



**Figure 3-14** File data sources

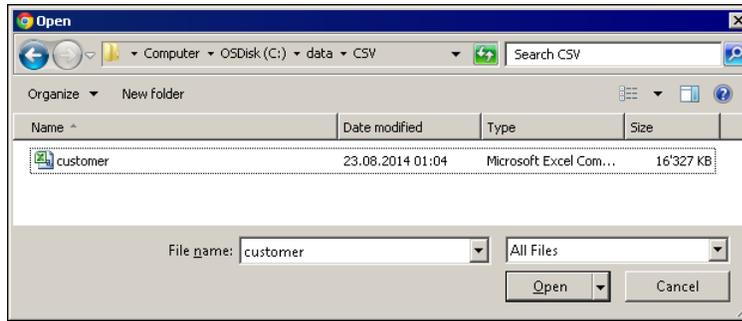
To access your text files you must know the file name and location.

## Loading local files

After selecting the file from your local file system, it uploads into the data folder of your load project. There are no file size limitations. See “Creating a load transaction” for more information about character encoding and end of line methods if you are unsure how to identify these values in your file.

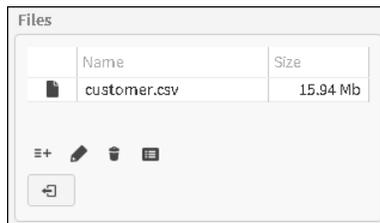
### How to upload a file from a local file system

- 1 In the *Files* panel, choose the upload icon. The *File Upload* screen from your operating system appears, as shown in Figure 3-15.



**Figure 3-15** Uploading a local file

- 2 In *File Upload*, navigate to the folder where the file is located.
- 3 Select the file and choose *Open*. The file appears in the *Files* panel, as shown in Figure 3-16. The local file is stored in the data folder of your project.



**Figure 3-16** Creating a local file data source

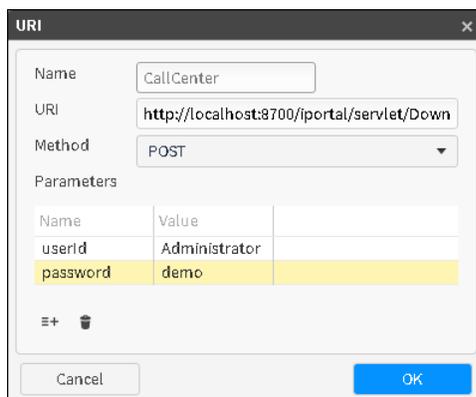
## Loading remote files

The *Remote data provider* option enables you to load CSV files using a Uniform Resource Identifier (URI) such as a web address. After selecting the file transfer protocol such as FTP or HTTP, you select the transfer method. The transfer method options are GET and PUT.

For example, to load a CSV file from a BIRT iHub server you POST a URL request for the file such as the following:

```
http://localhost:8700/iportal/servlet/DownloadFile?name=/Resources/data/callcenterdata.csv
```

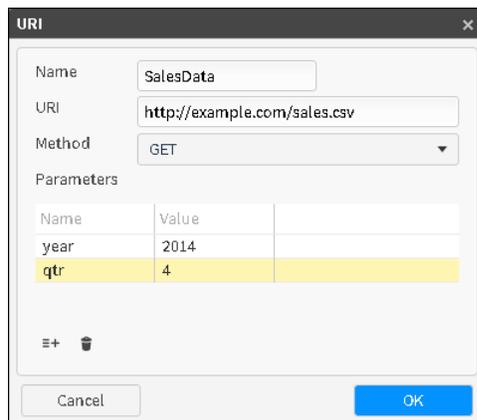
Add parameters for the user authentication as parameters. Figure 3-17 shows a URI to download a CSV file stored in a BIRT iHub server. You can add additional parameters and values that your remote data provider supports, such as location and date for a web site that offers weather information.



**Figure 3-17** Loading a CSV file from a server using HTTP

### How to load a file using the URI screen

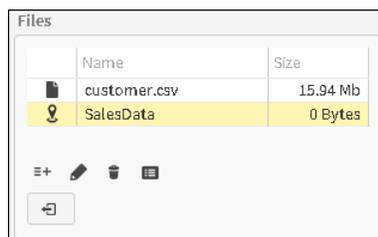
-  1 In *Files*, select the *Create a new remote data provider* icon. The URI screen appears.
-  2 In URI, type the following information, as shown in Figure 3-18:
  - In *Name*, type the name you want to assign to this data source.
  - In *URI*, type the URL that specifies the file location or the URL that generates the file.
  - In *Method*, select the method for obtaining the file. Contact the file provider to identify the method.
  - In *Parameters*, type any required parameters. Some web services support selecting data using parameters such as a location name or date.
    -  □ To add a parameter, choose *Add new parameter* icon again, near the bottom of the *URI* screen.
    - In *Name*, type the parameter name. In *Value*, type the parameter value, as shown in Figure 3-18.
    - To modify a parameter, select a column and change the entry.
    -  □ To remove a parameter, Click *Trash* next to the *Create a new remote data provider* icon.



Name	Value
year	2014
qtr	4

**Figure 3-18** Providing URI properties

- 3 Choose *OK*. The new remote file data source appears in the *Files* panel, as shown in Figure 3-19.



Name	Size
customer.csv	15.94 Mb
SalesData	0 Bytes

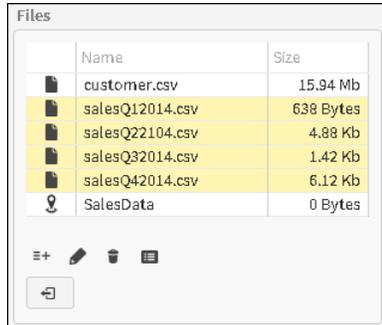
**Figure 3-19** Reviewing remote file data sources

### Loading files using a file list

On many occasions, due to a large data volume, it is convenient to break the data into multiple files. In such cases you must use a file list to create a file data source and upload the data into a single table.

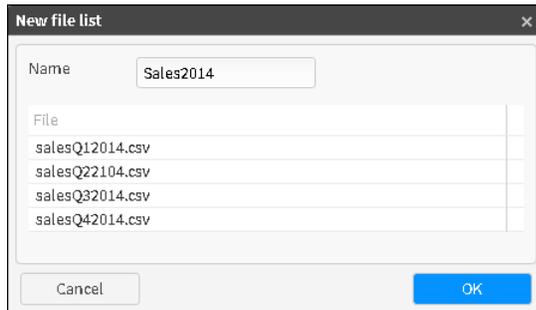
### How to load files using a file list

- 1 In the *Files* panel, upload all the files you want to include in a list, the same way you upload local files.
- 2 Press and hold the *Ctrl* key while using the mouse to select multiple file names, as shown in Figure 3-20.



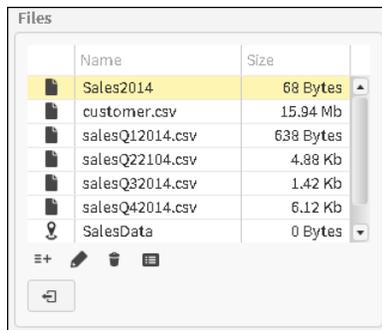
**Figure 3-20** Selecting multiple data source files

- 3 In the *Files* panel, choose the *New file list* icon at the bottom of the panel. The File names that you selected in step 2 appear, as shown in Figure 3-21.



**Figure 3-21** Naming a file list

- 4 In the *Name* field, type a name for data source of the new list file, then choose *OK*. The list file data source appears in the *Files* panel, as shown in Figure 3-22.



**Figure 3-22** Reviewing file data sources

---

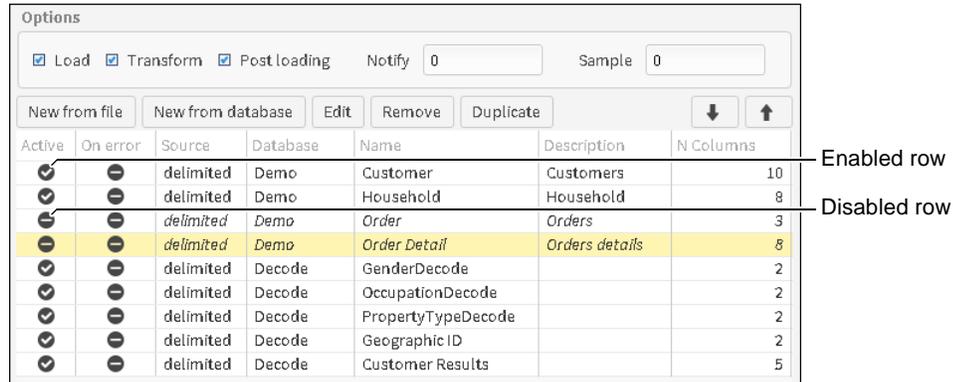
## Creating a load process

BIRT Analytics Loader supports creating data load transactions, setting load process options, and running a set of transactions as a process. To create a load process, first create the transactions that load data. Then, set options that run the load process.

## About transactions

A transaction includes all settings and data mappings required to load data into a single table. A load process executes multiple transactions. You must save any change that you make to a load process before executing the process. To save any change to a process, choose *Save* on the menu bar.

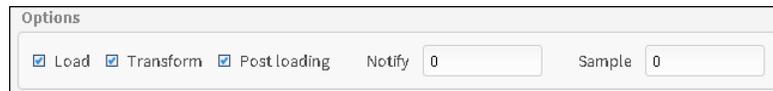
The *Options* panel in the *Data* tab provides options for defining a process and tools for creating transactions, as shown in the example in Figure 3-23.



**Figure 3-23** Defining data

## Setting load process options

Select the options that define your load process in the *Options* panel, shown in Figure 3-24. The settings that you save for each process execute when the process runs, as part of a data load project.



**Figure 3-24** Reviewing loading options

Select or specify values for each option to execute the following process tasks:

- **Load**  
Select Load to execute all load transactions when a project runs.
- **Transform**  
Select Transform to execute all transformation scripts when a project runs. Define transformation rules using the Transformation tab.
- **Post loading**  
Select Post loading to automatically synchronize the database and remove the cache.
- **Notify**  
Type a value that sets a trigger to write a log message each time the set number of records loads.
- **Sample**  
Specify the records loaded for each table.

## Ignoring items during the load process

Ignore items such as tables and instructions during the loading process using the *Data* tab and the *Transform* tab. Clicking on the “check” symbol in the *Active* column in the *Data* tab, as

shown in Figure 3-24 to ignore the element during the loading process (toggling them on and off). The contents of ignored item rows appear in italics.

---

## Creating a load transaction

Creating a load transaction involves defining the mappings between the source and target data objects. Mapping defines the relationship and data flow between source and target objects.

Before building an actual data map for the loading process, define a data model for the target even if it is not an actual database. This helps understand the entities it belong to and makes it easier to find the similar entity in the source and map it to the target element. This is called building a logical data map model and you have to do it before you actually start mapping the data.

After you have built a data map model for Execute, Transform, Load (ETL), you have to first analyze the data content, complete business rules for ETL process, integrate data sources to a single source, before you can build an actual data map.

BIRT Analytics Loader supports creating multiple load transactions per project.

You have the following choices when you create a load transaction:

- New from file  
Define a transaction that creates a table mapped to a flat file connection.
- New from database  
Define a transaction that creates a table mapped to a database.
- Edit  
Edit a load transaction.
- Remove  
Remove a load transaction.
- Duplicate  
Create a new transaction using an existing one.

## Mapping data from files

The mapping process requires defining the source and the target data. The software enables you to validate the data connection and create the table column definitions. You can automatically import the target table structure from the source file or manually create the column definitions. When the table structure does not map the source file data precisely, the loading process cannot identify the data to load for each table column and finishes unsuccessfully.

To map data from files you need to complete the following procedures:

- Defining a target database
- Defining a source file format
- Defining table columns

See the example in “How to map data from a CSV file”, found later in this chapter, for more information about the mapping process.

## Defining a target database

To define the target database table and the way the data is loaded, you need to open the *Table Declaration* screen by clicking on the *New from file* button in the *Options* panel of the *Data* tab.

Fill in the fields of the *Table declaration* screen as shown in Figure 3-25:

Name	Type	Width	Decimals	Format	Index	Skip
Cust ID	integer	18	0		✓	✗
DOB	date	8	0	yyyymmdd	✓	✗
Gender	string	1	0		✓	✗
Household ID	integer	18	0		✓	✗

**Figure 3-25** Creating a table mapping to a file

- **Action**  
For complete loads, choose “Create”, for incremental loads choose “Append”.
- **Database**  
Type a name for the target database, or select a listed database name.
- **Fail on error**  
Stop processing the table if there is an error. This is the same as the On error parameter when the table is viewed in the Data tab.
- **Ignore**  
Disable processing of the table when this option is enabled. This is the same as the Action parameter when the table is viewed in the Data tab or the Transform tab.
- **Description**  
Optionally, type characters that describe the target data table.
- **Name**  
Type a unique name for a target data table.
- **Overwrite**  
Replace an existing table with the same name with the new table.

### Defining a source file format

Based on your knowledge of source file properties, select formatting properties that most closely match those of the source file. More accurately selecting source file properties minimizes manual definition of target table properties.

- **Type**  
Select the file type. The files can be flat, or delimited. The flat files do not have delimiters and qualifiers.
- **Encoding**

Select an encoding type from the list. Table 3-2 lists and describes the file encoding types that BIRT Analytics supports.

**Table 3-2** Supported encoding

Encoding	Description
ASCII	American Standard Code for Information Interchange
CP1252	Windows Latin-1
BIG5	Chinese character encoding method used in Taiwan, Hong Kong, and Macau for Traditional Chinese characters
BIG5-HKSCS	Standard supplementary character set used for information exchange in Chinese between the public and the Hong Kong SAR Government
CP1252	Windows Latin-1
GB18030	Chinese government standard describing the required language and character support necessary for software in China
GB2312	Official character set of the People's Republic of China, used for simplified Chinese characters
GBK	An extension of the GB2312 character set for simplified Chinese characters, used in the People's Republic of China
ISO-8859-1	ISO 8859-1, Latin Alphabet No. 1
ISO-8859-2	Latin Alphabet No. 2
ISO-8859-3	Latin Alphabet No. 3
ISO-8859-4	Latin Alphabet No. 4
ISO-8859-5	Latin/Cyrillic Alphabet
ISO-8859-6	Latin/Arabic Alphabet
ISO-8859-7	Latin/Greek Alphabet
ISO-8859-8	Latin/Hebrew Alphabet
ISO-8859-9	Latin Alphabet No. 5
ISO-8859-13	Latin Alphabet No. 7
ISO-8859-15	Latin Alphabet No. 9
UTF8	8-bit UCS Transformation Format
UTF16	16-bit UCS Transformation Format, byte order identified by an optional byte-order mark
UTF16LE	16-bit Unicode Transformation Format, little-endian byte order
UTF16BE	16-bit Unicode Transformation Format, big-endian byte order
WINDOWS-1252	Windows Latin-1

- End Line

Select an end-of-line character from the list. Table 3-3 lists and describes the end-of-line characters and results supported by BIRT Analytics Loader.

**Table 3-3** End-of-line characters and results

End-of-line character	Result description
CRLF	Carriage return and linefeed
CR	Carriage return only
LF	Line feed only

- **File**  
Select a source file name from the list.
  - **Delimiter**  
Select the character used to delimit or separate values in a delimited file only. The following delimiter characters are supported by BIRT Analytics Loader:  
 , Tab | : ; @ # " + - = ~ '
  - **Header**  
Select Header option if the source file includes information in a header. A file header contains multiple column names, each separated by a delimiter character.
  - **Skip**  
In Skip, type a number that sets how many blank rows to insert between header information and record information in a file. If you select Header option, 1 appears in Skip by default.
  - **Offset**  
Define the initial position of each column when the file format is flat file.
  - **Qualifier**  
Select a qualifier character. BIRT Analytics Loader supports the following qualifier characters:  
 " ' ~  
 Choose None, if the file format does not use a qualifier.
  - **Well-formed**  
Select the Wellformed option to enforce verification that each record contains all columns.  
 Wellformed explanation:
    - *Enabled*- indicates the same number of columns in data source and table definition
    - *Disabled*- means that extra columns are ignored and Null values are set for missing columns
  - **View**  
Shows the data in the file and verify the connectivity to the file.
  - **Get columns**  
Choose *Get columns* to get the column names and type, extracting values from the source file.
- Use a text editor to verify parameters such as the character encoding and end of line method in your file if you are unsure of these values. You can view this information by opening your files with a text editor such as Notepad++. Some text editors do not support large text files, such as a multi-gigabyte database dump file. The following text editors are reported to support large text files:
- Gvim
  - Emeditor
  - 010 Editor
  - EditPadPro
  - Less

## Defining table columns

To automate defining the structure of a target data table, extract data from a source data file using the *Get columns* option in the *Table declaration* screen. During extraction, the *Get columns*

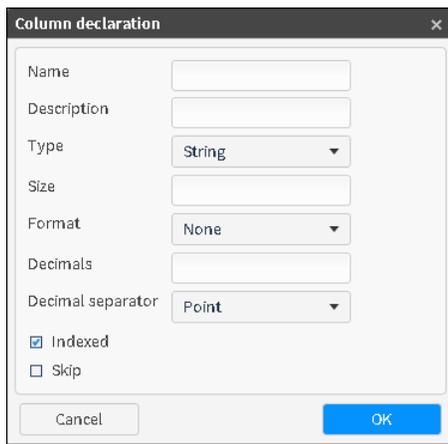
option adds column type and index information to each data element. You must review and edit, as necessary, the type and index information for every column. Try using *View* before using *Get columns* to verify that data is being read correctly from the file.

Consider removing the index from every column not strictly requiring an index. Removing unnecessary indexes speeds data load and analysis processes, but can limit performing aggregation and link operations. These operations require indexed columns.

To create a target data table manually, define each column using the *New*, *Edit*, *Remove*, and *Duplicate* options.

While you edit the structure of a target data table, use *Get columns* to retrieve existing column information from the source file, then use *Edit* to make modifications.

Create a new column by choosing *New* in the *Table declaration* screen, then provide information in *Column Declaration* screen that opens, as shown in Figure 3-26. Using *Get columns* overwrites any existing column definitions.



**Figure 3-26** Creating a new column

- **New**
  - Creates a new column in the target table
  - **Name**  
Enter the name of the column.
  - **Description**  
Enter a description of the column.
  - **Type**  
Choose the data type from the drop-down list in the “Type” field. The supported data types are described in Table 3-4.

**Table 3-4** Data types

Type	Description
String	A sequence of characters, up to 1024.
Integer	Integer (whole number) data from $-2^{31}+1$ (-2,147,483,647) through $2^{31} - 1$ (2,147,483,647).
Longint	Integer (whole number) data from $-2^{63}+1$ (-9,223,372,036,854,775,807) through $2^{63}-1$ (9,223,372,036,854,775,807).
Real	Floating precision number data with the following valid values: $-1.79769 \times 10^{308}$ through $1.79769 \times 10^{308}$ .

**Table 3-4** Data types (continued)

Type	Description
Date	The default format is mm_dd_yyyy, where the date separator can be slash (/), comma (,), period (.), or no separator.
Datetime	Date and time data from January 1, 1400, through December 31, 9999, with an accuracy of three-hundredths of a second, or 3.33 milliseconds. The default format is mm_dd_yyyy_hh_MM_ss.
Time	The default format is hhMMss.
Unicode	A sequence of Unicode characters, up to 1024.

*(continues)*

- Size  
The size of the column in characters, or digits.
- Format  
The format of the columns of type Date, Datetime, and Time.
- Decimals  
The number of digits to the right of the decimal point, or comma. Applies for data of type Real.
- Decimal separator  
You can choose Point or Comma to visualize a decimal separator. Applies for data of type Real.
- Indexed  
Indicates if the column is an index.
- Skip  
Select Skip to exclude a new column from the load process.
- Edit  
Opens the Table Declaration editor for a single column.
- Remove  
Removes a column from the target table.
- Duplicate  
Creates a new column by copying the properties of an existing column.

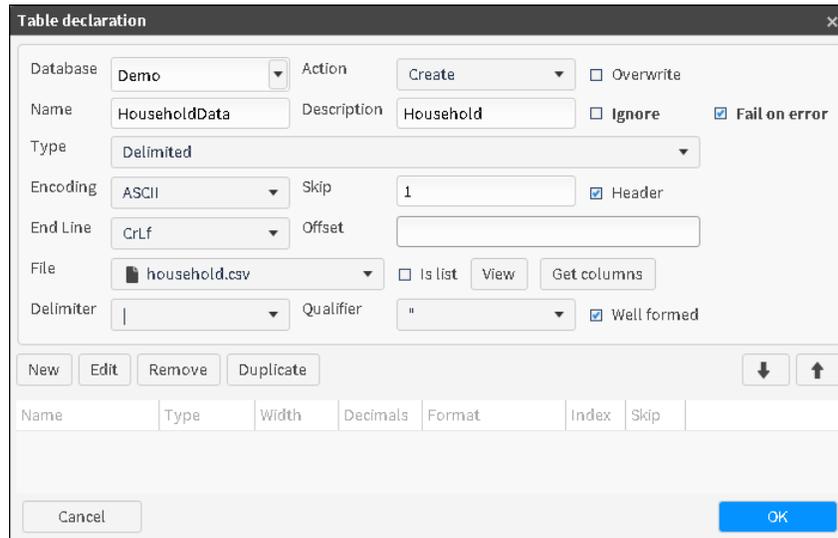
**How to map data from a CSV file**

The following procedure assumes that you have created a data file connection. It uses as an example, a CSV file having the following structure:

```
Address Line|Country|County|Household ID|Postal Code|Property Type|Region|
Town
P.O. Box 137, 8886 Ut Street|US|Los Angeles|1000|90275|T|Southern
California|Rancho Palos Verdes
Ap #687-3359 Eu Road|US|San Francisco|1001|94199|T|Southern California|San
Francisco
Ap #839-5476 Eros. Ave|US|Placer|1002|95604|T|Northern California|Auburn
```

- 1 In *Data-Options*, choose *New from file*.
- 2 In *Table Declaration*, fill in the fields to set properties for the target database, as shown in Figure 3-27.

- 1 In *Database*, type a database name. Alternatively, select an existing database name from the list of target databases.
- 2 In *Name*, type the name of the target table that receives the mapped data. For example, HouseholdData.
- 3 In *Description*, type information about the table contents, such as Household.
- 4 In *Action*, select “Create”.
- 5 Enable “Overwrite”.



**Figure 3-27** Defining file structure

- 3 Select options that match properties of the source file, as shown in Figure 3-27.
  - 1 In *Type*, select Delimited
  - 2 In *Encoding*, select ASCII
  - 3 Enable *Header*
  - 4 Edit the value in *Skip* to match the header rows in the source data file
  - 5 In *End Line*, select CrLf
  - 6 In *File*, select a source file name from the list
  - 7 In *Delimiter*, select pipe (|)
  - 8 In *Qualifier*, select "
  - 9 Enable *Well-formed*
- 4 Choose *View* to validate the table declaration. The file content appears, as shown in Figure 3-28. Choose *OK*.

Address Line	Coun...	County	Household ID	Postal Code	Prop...	Region	Town
P.O. Box 13...	US	Los Angeles	1000	90275	T	Southern California	Rancho Pal...
Ap #687-33...	US	San Francisco	1001	94199	T	Southern California	San Francisco
Ap #839-54...	US	Placer	1002	95604	T	Northern California	Auburn
P.O. Box 94...	US	San Joaquin	1006	95297	S	Northern California	Stockton
7945 Habita...	US	San Francisco	1008	94199	S	Northern California	San Francisco
6637 Ut Rd.	US	Santa Clara	1009	95196	B	Northern California	San Jose

**Figure 3-28** Viewing file content

5 Choose *Get columns* in *Table declaration* to create column metadata.

During the *metadata import* process, the *Get columns* option reads column names from the header row in the source file. It also determines a type for each column and indexes the column.

Definitions for each detected column appear as shown in Figure 3-29.

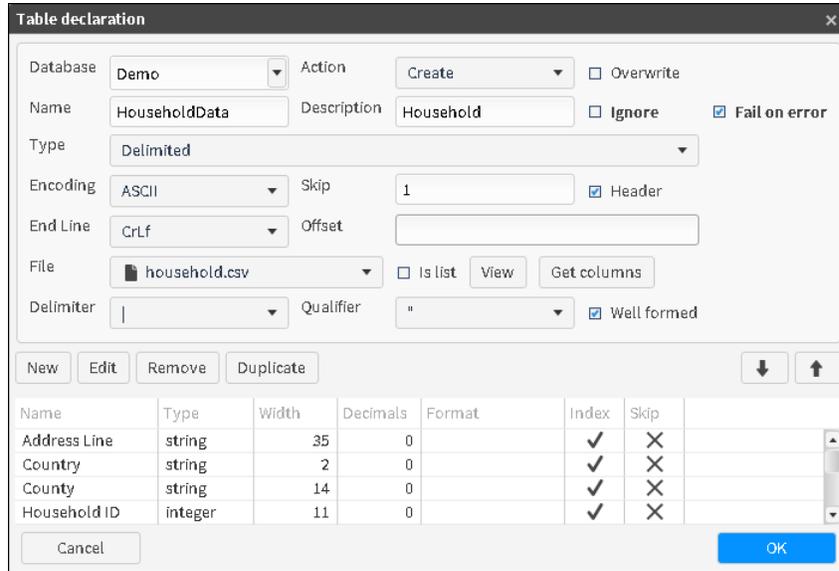


Figure 3-29 Getting column definitions

6 To modify column properties, double-click the column or select the column definition and choose *Edit*. This opens *Column declaration* where you can change the definition of the column and the following features:

- To speed data loading and analysis, you can remove column index information by removing the selection for *Indexed*. Indexing is required if the column is used in aggregation and link operations.
- To exclude a column from loading, select *Skip*.

Choose *OK*.

In this example, modify the column properties for the selected column, as shown in Figure 3-30.

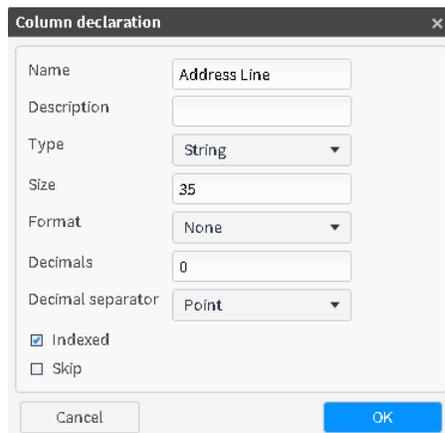


Figure 3-30 Editing column definition properties

7 Choose *OK* in *Table Declaration* to close the table declaration.

- 8 In Loader, choose *Save* to save the current project.

## Mapping data from databases

Mapping data from a database requires that you define many of the same properties as those that you define for mapping data from files. To map data from a database, first select the target FastDB database, provide a target table name, and select a data source. You can select a data source from the data base connections you have created.

Next, define the data set to retrieve from the source database using the *Table Declaration* screen that appeared when you clicked on the *New from database* option. This is described in more detail in the *Defining a data set* section.

Selecting a database in the *Datasource* drop-down list, provides the option to define a query that retrieves a data set. This is described in more detail in the “*How to map data from a PostgreSQL database*” section.

### Defining a data set

To define a data set you must provide a query and set options for the query.



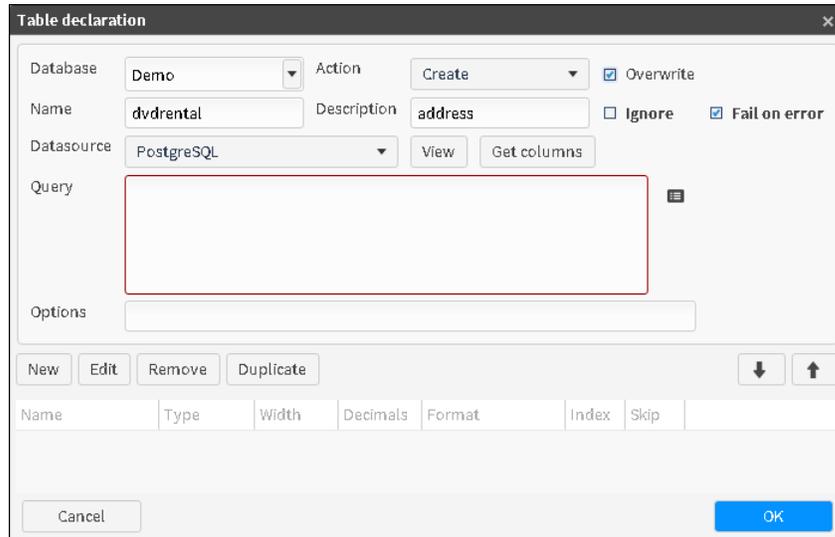
- **Query**  
Defines a SQL query from a relational database. You can type a query manually, and use the *Get columns* option to retrieve the columns from the data set.  
  
Alternatively, you can use the *View* option to select tables and columns. Use the drop-down list in the *Datasource* field to select tables and their columns, and specify the column order using the *up* and *down* arrows.
- **Options**  
Enter the options required by the connection drivers. For example, when loading data from an Oracle database, consider including a parameter, such as `PreFetchRows=10000`.

The process of defining the target table structure is the same as the one discussed in “*Defining table columns*,” earlier in this chapter. While editing the structure of a target data table, use the *Get columns* option to retrieve existing column information, then use *Edit* to make modifications. To create a new column, choose *New*, then provide information in the *Table Declaration* screen, as shown in Figure 3-31.

#### How to map data from a PostgreSQL database

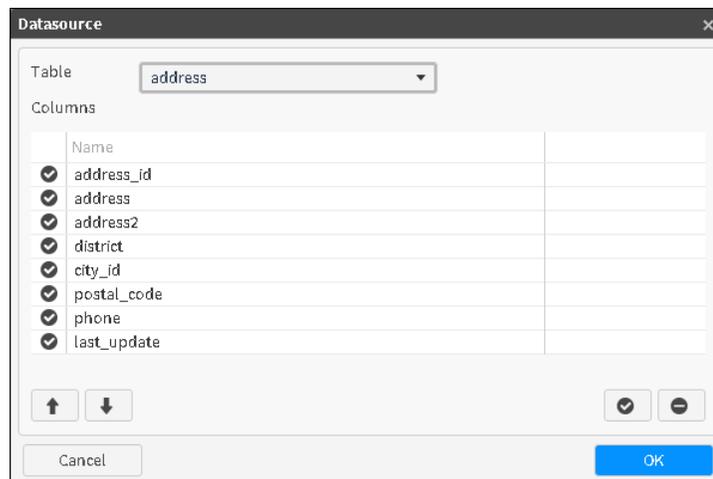
This procedure assumes you have already created an PostgreSQL database connection.

- 1 Define the target FastDB database and table.
  - 1 In the *Data* tab of your project, choose the *New from database* option to open the corresponding *Table Declaration* as shown in Figure 3-31. Choose the target database from the drop-down list in the *Database* field.



**Figure 3-31** Mapping data from a PostgreSQL database

- 2 Choose "Create" in the *Action* field.
  - 3 In *Name*, specify the name of the target table.
  - 4 In *Description*, type the table description.
  - 5 Select the *Overwrite* box
- 2** Define the *Query* to extract data.
- 1 In *Table declaration - Datasource*, select a database profile.
  - 2 Choose *View database structure* beside the *Query* field. The *Datasource* screen appears.
    - In *Datasource*, select a table from the drop-down list. The column names appear, as shown in Figure 3-32.
    - Use + and - to add or remove columns, and the up and down arrows if you need to change the column order.
    - In *Datasource*, choose *OK* to confirm the column definitions and generate the query.



**Figure 3-32** Selecting columns from a database table

This example generates the following query:

```
SELECT * FROM "address"
```

- In *Options*, specify any options you want to pass to the data driver.
- Choose *View* to preview the data connection and view the data. Your query results appear, as shown in Figure 3-33.

address_id	address	address2	district	city_id	postal_code	phone	last_update
1	47 MySakila Drive		Alberta	300			2006-02-15T09:45:30
2	28 MySQL Boulevard		QLD	576			2006-02-15T09:45:30
3	23 Workhaven Lane		Alberta	300		14033335568	2006-02-15T09:45:30
4	1411 Lillydale Drive		QLD	576		6172235589	2006-02-15T09:45:30
5	1913 Hanoi Way		Nagasaki	463	35200	28303384290	2006-02-15T09:45:30
6	1121 Loja Avenue		California	449	17886	838635286649	2006-02-15T09:45:30

**Figure 3-33** Viewing data from a database

- Choose *Get columns* to retrieve the columns from the target table. The column definitions appear as shown in Figure 3-34. The types are determined automatically, and all columns are indexed by default. Get columns overwrites the existing data set, and discards your existing changes.

Name	Type	Width	Decimals	Format	Index	Skip
address_id	longint	10	0		✓	✗
address	unicode	50	0		✓	✗
address2	unicode	50	0		✓	✗
district	unicode	20	0		✓	✗
city_id	integer	5	0		✓	✗
postal_code	unicode	10	0		✓	✗
phone	unicode	20	0		✓	✗
last_update	datetime	23	0	yyyy_mm_dd_...	✓	✗

**Figure 3-34** Getting the column definitions from the data set

- Edit the column declaration.

The example in Figure 3-35 changes the format of the `last_update` column, and disables indexing of the column.

Column declaration

Name: last\_update

Description:

Type: Date

Size: 23

Format: mm\_dd\_yyyy

Decimals: 0

Decimal separator: Point

Indexed

Skip

**Figure 3-35** Editing a Datetime column

- Select the `last_update` column and choose *Edit*.
- In *Type*, select *Date* format.
- In *Format*, select `mm_dd_yyyy`.

4 Disable *Indexed* and choose *OK*.

6 Choose *Save* to save the project.

## Managing errors in the loading process

In *Data - Options* you can decide to continue the loading process when errors occur.

By default, the loading process stops whenever an error is encountered. You can continue loading data when error occur using *On Error*. This enables the loading process to ignore any tables that generate errors during the process, leaving them to be fixed and loaded later if desired.

Ignoring errors is useful when loading high volume tables overnight.

### ON ERROR Attribute

In *Data - Options*, choosing the “minus” symbol (in the “On error” column) beside a table row, changes the “minus” symbol to an “arrow” symbol and deactivates the default “Fail on error” setting in the list of tables to be loaded, as shown in Figure 3-36.

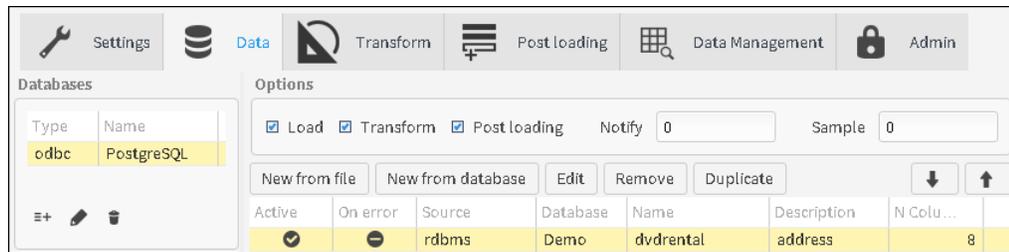


Figure 3-36 Reviewing new table entries

Double clicking on the table opens its *Table declaration* window where you can see that the *Fail on error* check-box is now “unchecked”. Clicking the “arrow” symbol changes it back to “minus” and resets the “Fail on error” setting, selecting the box again as shown in Figure 3-37.

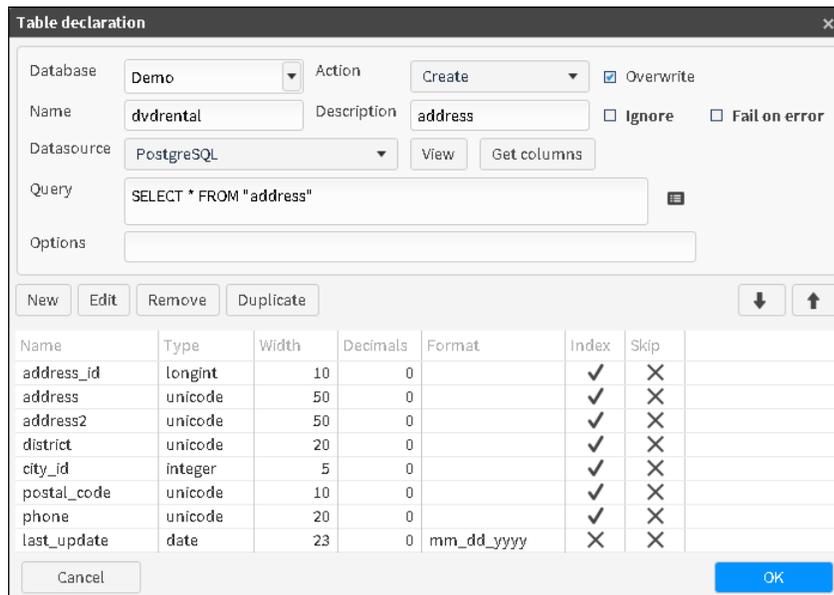


Figure 3-37 Managing exceptions using the Data tab



# 4

## Managing data

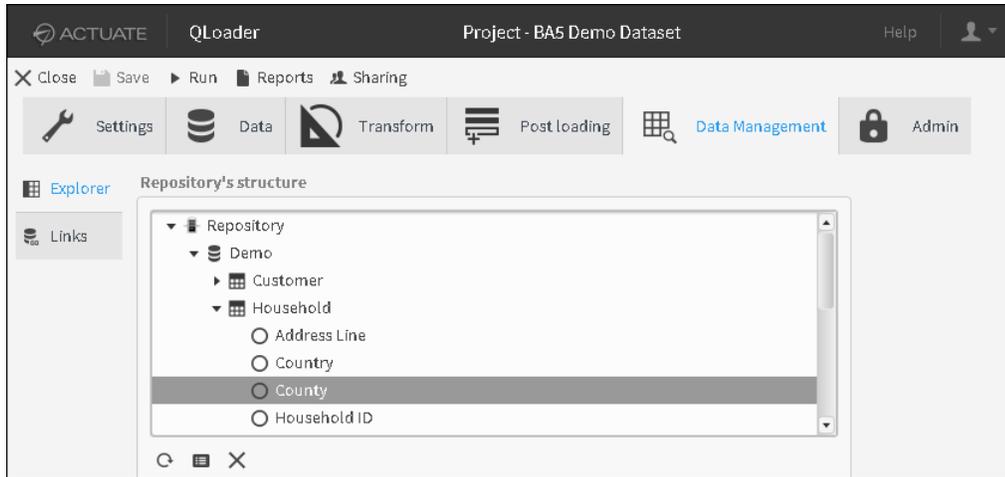
This chapter contains:

- About managing data
- Understanding Explorer
- Understanding dynamic and permanent columns
- Understanding Links

---

## About managing data

The main screen of the BIRT Analytics Loader Data Management tab, shown in Figure 4-1, provides tools for exploring data, creating, and maintaining links between tables. BIRT Analytics Loader connects directly to the database engine.



**Figure 4-1** Selecting an available data table

The Data Management tab opens with the *Explorer* tab, where you can explore the structure of all databases currently stored in FastDB. This panel shows data objects, such as databases, tables, columns and their properties including the discrete values of each column.

The Data Management *Links* tab provides a space for creating and maintaining links between tables.

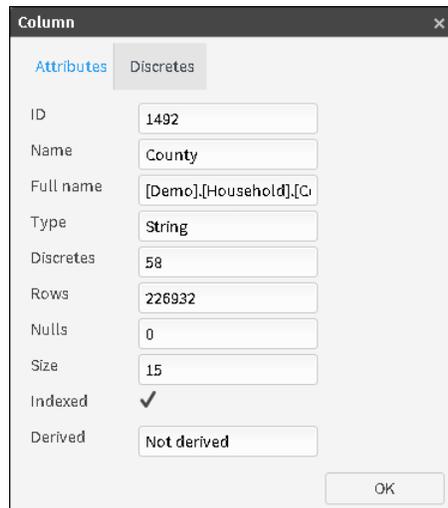
---

## Understanding Explorer

Use the *Explorer* tab to select and browse database structures stored in FastDB, such as databases, tables and columns.

### How to view data structures stored in FastDB

- 1 In BIRT Analytics Loader, choose *Data Management*. *Explorer* appears and shows the repository structure. The *Repository structure* panel lists all databases names stored in FastDB.
- 2 Expand a database name to view a list of all its tables.
- 3 Expand a table name to view a list of all its columns.
- 4 Select a column and choose View Details to view the column attributes and discrete values. See Figure 4-2.



**Figure 4-2** Viewing column attributes in the Data Management tab.

The three icons at the bottom left of the Explorer tab, shown in Figure 4-2, are used for:



- Refresh

In Repository, choose the Refresh icon to update your view of all databases, tables, and their columns currently stored in FastDB.



- Drop

To drop databases, tables, or data columns, select the data object in the Repository panel. Then choose the Drop icon, which removes the data object from FastDB.



- View details

To view properties of any repository objects, select the object and choose the View icon which shows the properties of selected objects.

## Viewing table properties

Select a table name in the Repository panel. Then choose *View* to view table properties. The following properties are displayed:

- ID  
Internal table identification number
- Name  
Table name
- Full name  
Full table name in format [Database].[Table]
- Description  
Table description
- Rows  
Number of rows in a table
- Columns  
Number of columns in a table

For example, Figure 4-3 shows all properties defined for the Household table in the Demo database.

Table	
ID	13
Name	Household
Full name	[Demo],[Household]
Description	Household
Rows	226932
Columns	12
OK	

**Figure 4-3** Exploring table properties

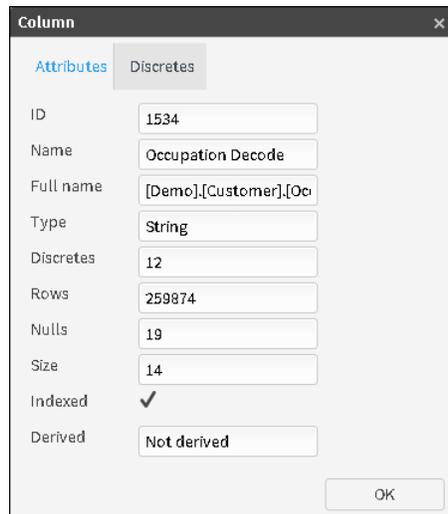
## Exploring column properties

To view column properties, in the Repository tree, double-click a column name to open the *Column* screen where you can view both the Attributes and Discretes for the chosen column.

The *Attributes* tab in the *Column* screen displays a value for each of the following column properties:

- **ID**  
Internal code that identifies the column
- **Name**  
A column name
- **Full name**  
Full column name in the following format:  
[Database] . [Table] . [Column]
- **Type**  
Type of data (text, integer, and so on)
- **Discretes**  
Number of column's discrete values
- **Rows**  
Number of rows in the database
- **Nulls**  
Number of column's null values
- **Size**  
Column size
- **Indexed**  
Indicates if column is indexed
- **Derived**  
Indicates if the column is dynamic or permanent

For example, Figure 4-4 shows all properties defined for the Occupation Decode column in the Customer table.

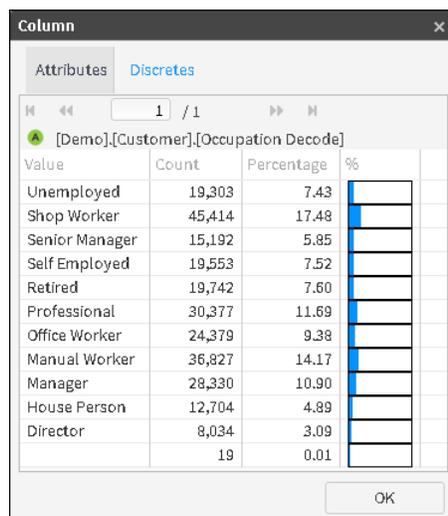


**Figure 4-4** Exploring properties of a permanent column

In this case the column is “Not derived” (dynamic - having no data values). *Discretizes* displays values for only the following column properties:

- Value  
Discrete value of the column
- Count  
Number of records for which the column has this value
- Percentage  
Percentile of the count number to the total number of records

For example, Figure 4-5 shows all discrete values defined for the *Occupation Decode* column in the *Customer* table.



**Figure 4-5** Exploring discrete values

---

## Understanding dynamic and permanent columns

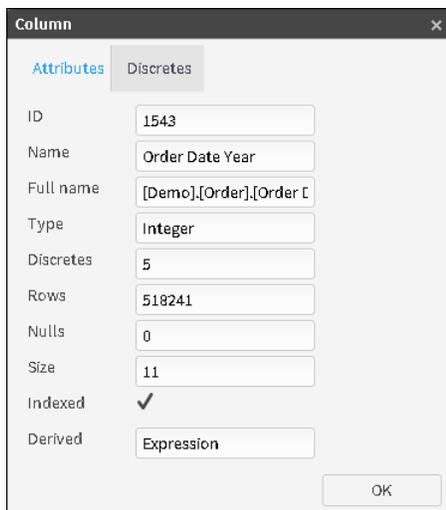
A *permanent column* contains data values. You typically load the data values in a permanent column from an external database.

BIRT Analytics uses the term *dynamic column* to describe a column that contains only a *formula*, but *no data values*. A formula consists of one or more expressions and operators.

Using BIRT Analytics Loader, you can edit the formula that defines a dynamic column. To edit such a formula, you change any of the expressions or operators in the formula.

You create a *dynamic column* in Loader using instructions known as *transformation instructions*. The dynamic columns appear as editable columns in the BIRT Analytics user interface.

For a *dynamic column*, the column attribute *Derived* displays the name of the transformation instruction that you use to create the column. For example, Figure 4-6 shows the column properties for a dynamic column named [Order Date Year]. It was created using the translation instruction, *Expression*, that appears in the Derived property field.



The screenshot shows a dialog box titled "Column" with two tabs: "Attributes" (selected) and "Discretets". The "Attributes" tab contains the following fields:

ID	1543
Name	Order Date Year
Full name	[Demo],[Order],[Order C
Type	Integer
Discretets	5
Rows	518241
Nulls	0
Size	11
Indexed	<input checked="" type="checkbox"/>
Derived	Expression

An "OK" button is located at the bottom right of the dialog box.

**Figure 4-6** Exploring properties for a derived column

You can transform a dynamic column to a permanent column using the transformation instruction, *MAKEPERMANENT*. For more information about transformation instructions, see, Chapter 5, "Transforming data."

---

## Understanding Links

You can link two tables together so records from one table are related to records in another table. Between two tables there can be only one link. When you link, you identify a column that is common to both tables. BIRT Analytics supports the following linking methods to match records in different tables:

- 1 to 1, associate a single record in one table to a single record in another table. For example, a table that gives customer names and customer id can link to a customer id in another table that gives additional details about each customer.
- 1 to N, associate a single record in one table to multiple records in another table. For example, a single customer can link to multiple orders.

The column in the table you are linking from must only have discrete values such as customer id numbers. The second table that you link to does not require discrete values. BIRT Analytics

sets the table with only discrete values as the parent table. If both tables contain only discrete values, then the table that is linked from becomes the parent table. The FastDB engine does not support linking from a column with continuous values, such as the time of day or a range of prices.

For example, if you use an Orders table and a Customers table, and you link a customer id column that exists in both tables, BIRT Analytics will set the Customers table as the parent table because all of the customer ids in the Customers table are unique.

Links with a 1 to N relationship, link a column indicated in the "From" field to the column indicated in the "To" field. Figure 4-7 shows a 1 to N relationship between two tables. BIRT Analytics sets the Demo.Household.Household ID column as the parent table because all of the Household ID values are unique in that table. The Demo.Customer.Household ID table is set as the child table because more than one customer can have the same household id.

**Figure 4-7** Populating FROM and TO fields when creating 1 to N links

Link types are shown in Data Management—Links. The following types are used in by the FastDB engine:

- Final
- Transient

Final links are the links you create using the Links option or the transformation instruction, LINK. For more information on LINK, see “LINK” in Chapter 5, “Transforming data.”

Transient links are automatically generated by the software for performance reasons.

For example, the transient link, shown in Figure 4-8 , gets created if you create a Crosstab with [Customer].[Occupation] as a dimension and Count( Order Detail ) as a measure.

Type	From	To
Final	[Demo].[Household].[Household ID]	[Demo].[Customer].[Household ID]
Final	[Demo].[Customer].[Cust ID]	[Demo].[Order].[Cust ID]
Final	[Demo].[Order].[Order No]	[Demo].[Order Detail].[Order No]
Transient	[Demo].[Customer]	[Demo].[Order Detail]

**Figure 4-8** Managing links

To clear the transient links, choose Remove all transients.

#### How to create a new link

- 1 Choose Links in Data Management.
- 2 In Link—From, select the database, table, and column where the link starts.
- 3 In Link—To, select the database, table, and column where the link goes, as shown in Figure 4-9.

The screenshot shows a dialog box titled "Link". It contains two rows of dropdown menus. The first row, labeled "From", has three dropdowns with the following values: "[Demo]", "[Order]", and "[Cust.ID]". The second row, labeled "To", has three dropdowns with the following values: "[Demo]", "[Customer]", and "[Cust.ID]". At the bottom left of the dialog, there are two buttons: "New" and "Remove".

**Figure 4-9** Creating links

- 4 Choose New to create the link.
- 5 Confirm your link creation, choose Yes when asked to confirm that you want to create a new link.

**How to remove a link**

- 1 In Link—From, select the database, table, and column where the link starts.
- 2 In Link—To, choose the database, table, and column where the link goes.
- 3 Choose Remove to drop the link.
- 4 Choose Yes when asked to confirm the change. The link disappears from the list of links.

# 5

## Transforming data

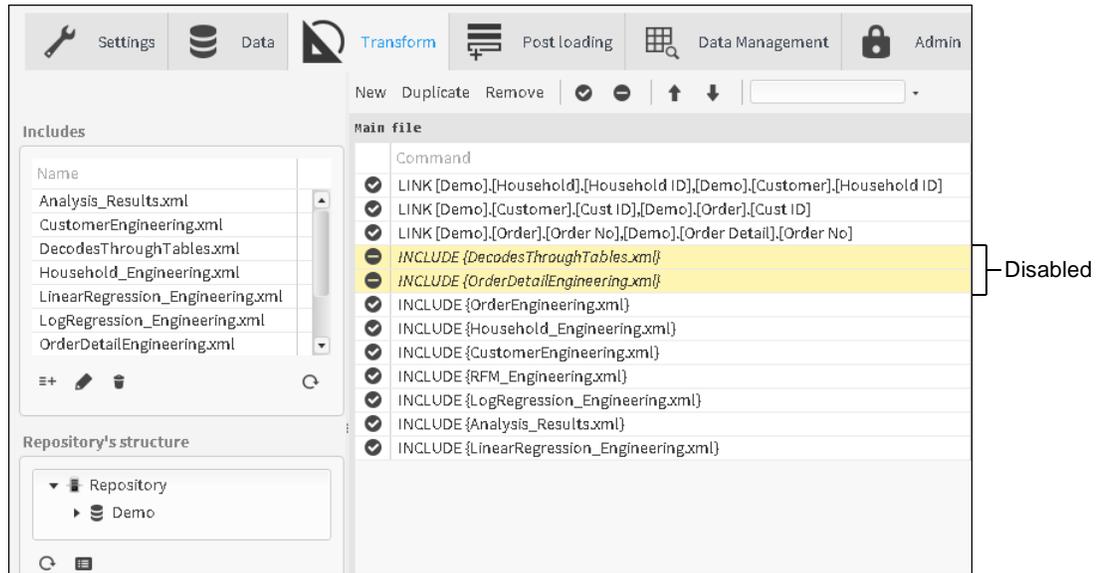
This chapter contains:

- About transforming data
- Creating new columns
- Managing data objects
- Miscellaneous instructions

## About transforming data

Transformation is the process of manipulating data. Any manipulation beyond copying is a transformation. Examples include cleaning, aggregating and integrating data from multiple sources. Transforming the data is executed after the load process is finished.

Transformation processes are managed in the *Transform* tab of BIRT Analytics Loader. Figure 5-1 shows the transforms used in the demo database.



**Figure 5-1** Transforming data

The *transformation* script consists of one or more *script* files. The list of the *script* files is displayed in the *Name* section of the *Includes* panel in the *Transform* tab. The *Main* file, displayed in the *Main file* panel, is the first file executed after a data load.

A new form for creating a new transaction opens on the right side of the Main file area. This happens whenever you choose a new Instruction type to define.

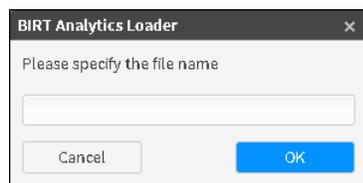
You can resize each of the three areas on the Transform screen by positioning your pointer on the area borders, and dragging the pointer in the desired direction.

## Understanding the Includes panel

Use the *Includes* panel to create script files. To execute these script files, you must include them in the *Main file* script.



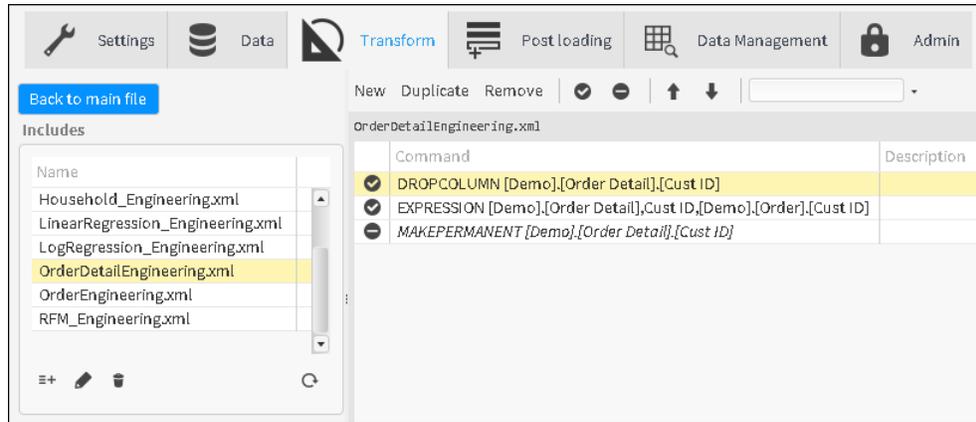
- Opens a new Include file  
Choosing Open requests a new file name, as shown in Figure 5-2.



**Figure 5-2** Creating a new script



- Edits a script file  
Selecting Edit opens the file in the editing script area as shown in Figure 5-3.



**Figure 5-3** Creating instructions for a new script

You can copy, paste, and delete instructions in Include files using the shortcut keys, shown in Table 5-1.

**Table 5-1** Editing shortcut keys

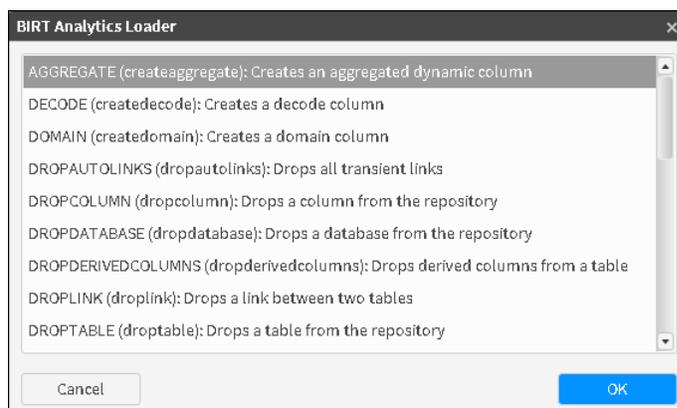
Functionality	All browsers shortcut keys	Firefox-only shortcut keys
Copy	Ctrl+Insert	Ctrl+C
Paste	Shift+Insert	Ctrl+V
Cut	Shift+Delete	Ctrl+X

-  ■ Removes a file
-  ■ Refreshes the file list in Includes

## Understanding scripts

The *Transform* tab opens showing the *Main file* panel where Transform scripts are displayed. Each script file contains transformation instructions. To create these instructions you use the following options at the top of the *Transform* tab:

- **New**  
Create a new instruction. When you choose “New” to create a new transformation instruction, the list of instructions appears, as shown in Figure 5-4. You select an instruction from the list, and provide the appropriate parameters for the instruction.



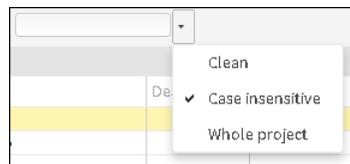
**Figure 5-4** Transformation instructions

Depending on their purpose, the transformation instructions are grouped in three categories:

- Instructions for *creating* new columns
- Instructions for *managing* data objects
- *Miscellaneous* instructions

For more information on transformation instructions, see the user guide *Using BIRT Analytics*.

- Duplicate  
Create a new instruction based on an existing one.
- Remove  
Remove an instruction.
-  ■ Enable  
Enable an instruction for execution.
-  ■ Disable  
Mark an instruction as not executable. You can use this to include comments in the script.
-  ■ Up arrow  
Move an instruction up in the script
-  ■ Down arrow  
Move an instruction down in the script
- Search field  
Empty field beside the arrow icons that enables a search for an instruction, as shown in Figure 5-5.

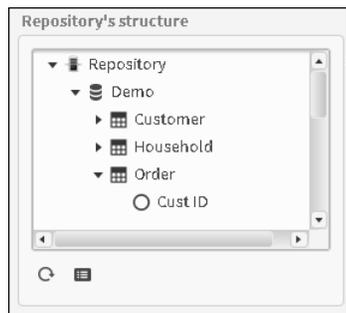


**Figure 5-5** Searching through transforms

You can copy, paste, and delete instructions using the shortcut keys described in Table 5-1.

## Understanding the repository structure

The Transform—*Repository's structure*, shown in Figure 5-6, shows the databases, their tables, and columns. Use this panel to drag the name of FastDB objects such as databases, tables and columns and drop them into transformation instructions. Double-click a column name to explore the column properties and discrete values.



**Figure 5-6** Repository's structure panel

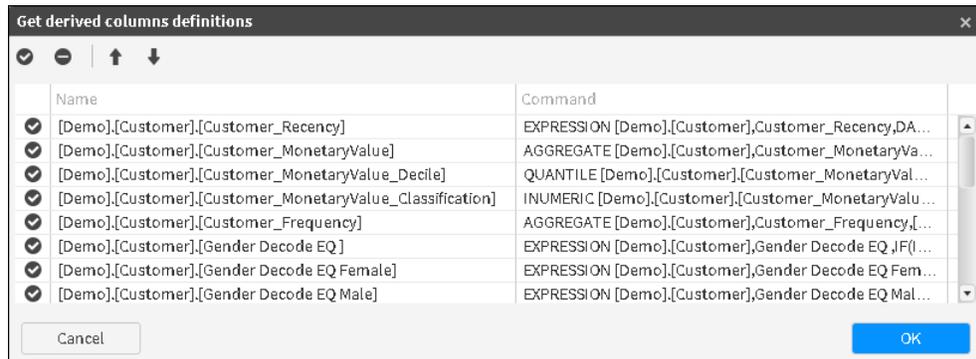


- Refresh the repository content.



- Get derived column definitions.

Displays a list of the derived columns in the selected database. These columns were added by analysts using Loader or BIRT Analytics and are not permanent.



**Figure 5-7** Getting a derived column definition

Analysts create derived (calculated) columns during the data analysis. BIRT Analytics Loader administrators can add these columns to the tables when refreshing the data during the load process. If derived columns are not made permanent, the definitions of these columns are lost, and must be recreated.

#### How to make a derived column permanent

Log in to Loader to complete the following instructions:



- 1 In Transform—Repository’s structure, choose Get derived column definitions.



- 2 In Get derived columns definitions, disable any columns that you do not want to become permanent.

- 3 Choose OK.

- 4 The new columns appear in the Transform—Main file list of loading instructions. The new permanent columns are included the next time you run the loading instructions.

## Using the text editor

You can invoke the text editor from any box requiring text entry. Double-click the text box and the text editor appears, as shown in Figure 5-8.



**Figure 5-8** Using the text editor

## Creating new columns

The instructions in this group create new dynamic columns in the database tables, using analytical functions to process the data. You can only use the following characters in column names: [ ] and " (also known as left square bracket, right square bracket, and double quotation mark). Once created the dynamic columns contain the formulas, used in their definition.

The formulas can be edited and so the values of the columns get changed. The dynamic columns get populated with data at run time when analysis operations are performed. You have an option to make the columns permanent, and then the formulas are replaced with real data.

## AGGREGATE

Creates an aggregated dynamic column. Aggregate enables you to group information from different tables. For example, you can group information about orders in a table that contains customer data to determine how many orders a customer has placed. Aggregation works with any table in the database as long as the tables are properly joined.

### Parameters

#### Target table

Defines the name of the target table, for example [SalesData].[MonthlySales]. The definition contains the database name, followed by the table name.

#### Target column

Defines the name of the target column. Do not use the following characters: [ ] and " (also known as left square bracket, right square bracket, and double quotation mark).

#### Source column

Defines the source column for the aggregation. The full definition includes, the database name, the table name, and the column name.

#### Function

Defines the aggregation function. The supported functions are shown in Table 5-2.

**Table 5-2** Aggregation functions

Function	Returns
Average	Average value
Count	Record count
First	Minimum value of an alphabetically sorted range of records
Last	Maximum value of an alphabetically sorted range of records
Sum	Sum of the column value in all records
Mean (integer)	Mean value in Integer format
Mean (real)	Mean values in Real format
Minimum value	Minimum value of the selected range of records
Maximum value	Maximum value of the selected range of records
Standard deviation	Value indicating the average distance from the average
Square sum	Sum of squares of the distance between each data point and the line of best fit

#### Filter

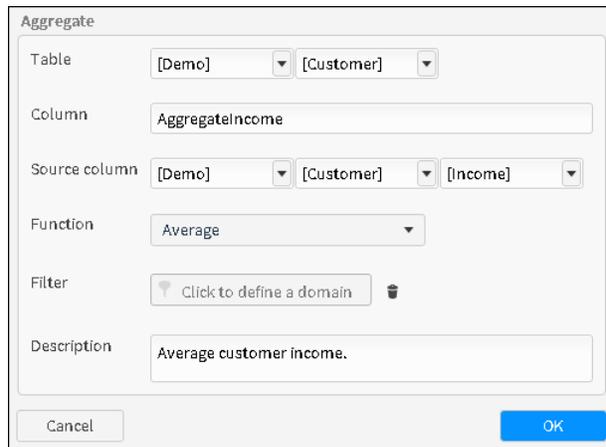
Use Domain definition to define a filter.

#### Description

Type text that describes the new column. You can type the text in the Description text box, or double-click to open a text editor and type the text there.

#### How to create an aggregating column

- 1 In the *Transform* tab, choose New. The list of transformation instructions appears.
- 2 In the list of transformation instructions, select AGGREGATE. Aggregate appears as shown in Figure 5-9.



**Figure 5-9** Creating an aggregating column

- 3 In *Table*, select the target database and table from the drop-down lists.
- 4 In *Column*, type the name of the target column.
- 5 In *Source column*, select the database, the table, and the column you want to use to create the new column.
- 6 In *Function*, select a function from the drop-down list.
- 7 In *Filter*, choose Click to select define a domain, to define a filter. Domain definition appears. For more details about defining a domain see DOMAIN, later in this chapter.
- 8 In *Description*, type text that describes the new column.
- 9 Choose “OK”. The aggregate instruction appears in the list of transformation commands, as shown in Figure 5-10.



**Figure 5-10** Creating a transformation instruction

- 10 Choose “Save” to save the instruction in the main file.

## DECODE

Creates a dynamic decoding column. You can decode any column. For example, gender is usually encoded as M and F, or even as 0 and 1. BIRT Analytics makes it possible to change the name to something more recognizable, such as Male and Female.

**Parameters** **Source column**  
Defines the source column name.

**Target column**  
Defines the target column name. The column is added to the source table. Do not use the following characters: [ ] and " (also known as left square bracket, right square bracket, and double quotation mark).

**File**  
The name of the file containing decoding values. The file format must comply with the text file formats supported by BIRT Analytics Loader. Use the *File Upload* icon in the *Files* panel of in the *Data* tab to upload the decoding file and make it available in BIRT Analytics Loader.

**Code position in file**

The position of a code in the file.

**Decode position in file**

The position of a decoding value in the file.

**Encoding**

Defines the file encoding.

**End line**

The end-of-line character. Select a character from the drop-down list.

**Delimiter**

Separates the values in the file. Select a delimiter from the list of supported characters.

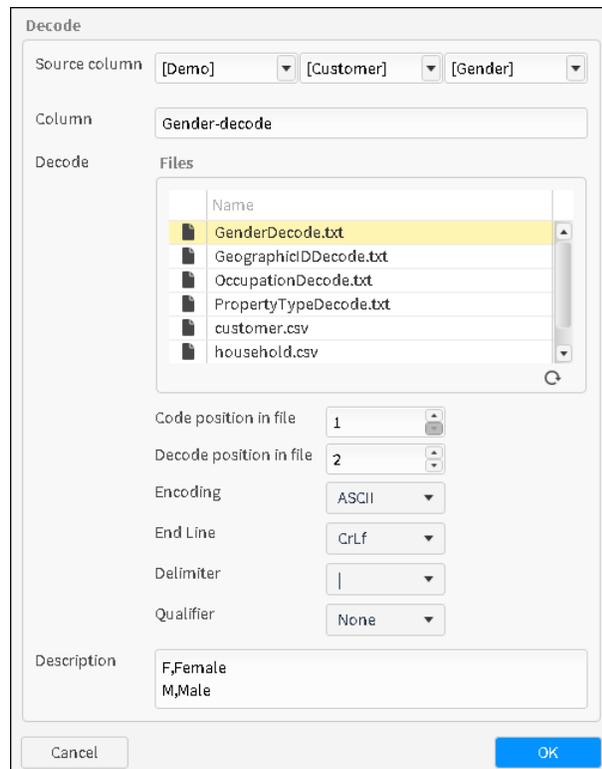
**Qualifier**

Surrounds the values in the file. Select a qualifier from the list of supported characters.

**How to create a decoding column**

Before creating a new decoding column, you must create a *text file* with the decoding values and deploy the file to BIRT Analytics Loader.

- 1 In the *Transform* tab, choose “New”. The list of transformation instructions appears.
- 2 In the list of transformation instructions, select DECODE. Decode appears, as shown in Figure 5-11.



**Figure 5-11** Creating a decoding column

- 3 In the *Source column* fields, select a database, table, and column name from the drop-down lists.
- 4 In the *Column* field, type a name for the new column.

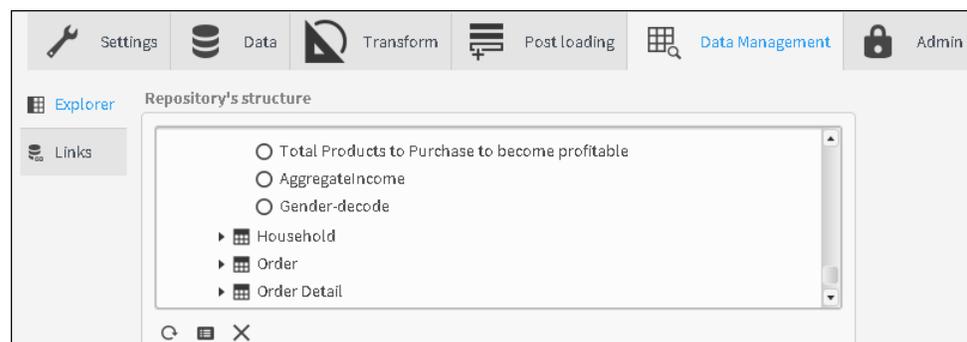
- 5 In the *Files* panel, select a file with decoding values. In this example, *decode-gender.txt* contains the following two comma-separated columns with the following coding and decoding values:
 

```
F, Female
M, Male
```
- 6 In the *Code position in file* field, type or select the number of the code column in the file. In this example, the position of the code is 1.
- 7 In the *Decode position in file* field, type or select the position of the decoding column. It is 2 in this example.
- 8 In the *End line* field, select the end line character used in the file from the drop-down list.
- 9 In the *Delimiter* field, select the delimiter used in the file from the drop-down-list.
- 10 In the *Qualifier* field, select the qualifier from the list.
- 11 In the *Description* panel, type a description of the column.
- 12 Click on the *OK* button at the bottom of the *Decode* screen. The *Decode* instruction appears in the list of transformation commands, as shown in Figure 5-12.

Command	Description
AGGREGATE [Demo].[Customer],AggregateIncome,[Demo].[Custo...	Average customer income.
DECODE [Demo].[Customer],[Gender],Gender-decode,{Transform...	F,Female M,Male

**Figure 5-12** Creating a Decode instruction

- 13 Click on the *Save* tool at the top of the *Loader Project* screen to save the project.
- 14 Run the project to validate the new column that was created.
  - 1 Go to the *Explorer* in the *Data Management* tab to select your database and expand the list of tables.
  - 2 Double-click the table you added the column to. The new decode column, in this example “Gender-decode”, appears in the list of columns as shown in Figure 5-13.



**Figure 5-13** Validating the decode column

## DOMAIN

A domain is a segment, or a set of records chosen for a specific purpose. A database is made up of records, and these records internally form groups that have similar features. For example, a company’s customers can be female or male, but not both. For example, to specify the target audience for a marketing campaign, you can combine segments, such as females at age between 25 and 35, from Barcelona, and average purchasing power.

A *Domain* instruction can be defined as a simple query, or as a more complex set of segments.

**Parameters** **Target table**  
 Defines a target table.

**Target column**  
 Defines a target column. Do not use the following characters: [ ] and " (also known as left square bracket, right square bracket, and double quotation mark).

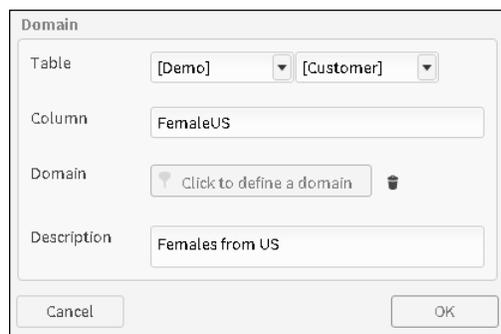
**Domain**  
 Defines a domain name. Specify a query that defines a domain, using any of the tools described in Table 5-3.

**Table 5-3** Domain definition features

Icon	Tool	Description
	Simple query	Defines a simple query. Allows you to create a query, by selecting database columns, and operations from the database.
	Block	Defines a new block of segments. A segment can contain multiple queries. A domain can be build of multiple blocks. The blocks act as parenthesis and define the order of execution.
	Select all	Select all records from a table.
	Resolution	Changes the resolution of the whole domain or of a set of values entered in a block. For example, if you make a domain of records of people who are women and managers, and you want to see the households where the people who meet these conditions live, you can use this option to verify the results in a dimension other than the source of the values.
	Inversion	Inverts the selected value or values. For example, if you use Invert on a domain that contains the records of people that are women and managers, you get the records of people who are not women and not managers.
	Discrete values	Selects the values that create the domain. Select a value from a column in the database.
	Sorting	Sorts the values in a column in ascending or descending order.
	Sample selection	Creates a sample of number of records.

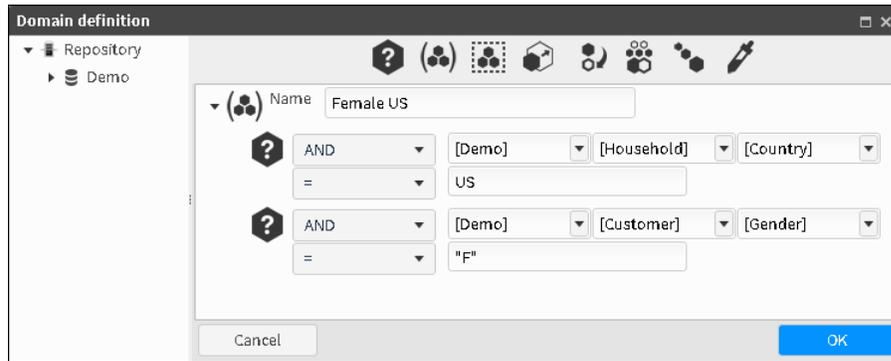
**How to create a domain**

- 1 In the *Transform* tab, choose New. Then choose DOMAIN in the list of transformation instructions. The *Domain* screen appears.
- 2 In the *Table* fields, select the database, and the table from the drop-down list, or select the table in the *Repository's structure* panel and drag it to the column box.
- 3 In the *Column* field, type the new column name, as shown in Figure 5-14.



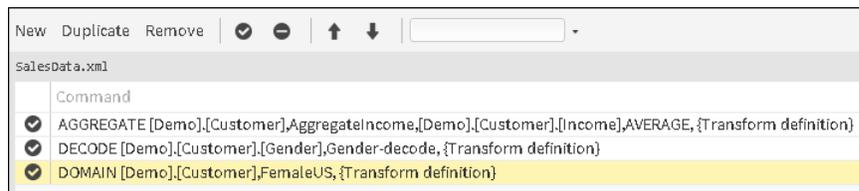
**Figure 5-14** Creating a domain

- 4 Click on the *Domain* field to open the *Domain definition* screen.
- 5 Create a new query, as shown in Figure 5-15.



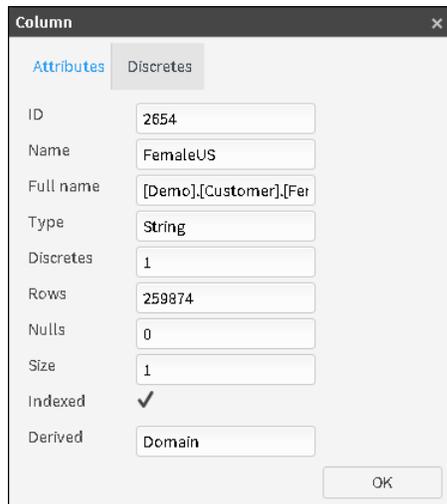
**Figure 5-15** Defining a domain

- 1 In *Name*, type the name of the domain.
  - 2 Drag *New query* from the toolbar over the *New block* icon, located on the left side of the *Name* field label, as shown in Figure 5-15.
  - 3 In *New query* fields, select a database, a table, and a column, or drag a column from Repository. Then select the operator, for example =, from the drop-down list.
  - 4 In the *New query* fields, type a value, for example US.
  - 5 Drag the *New query* icon over the *New block* icon to add a second condition.
  - 6 Select the *operator* between the two conditions. If you want to get the records for all women in US, you must select AND.
  - 7 Fill in all the Query fields for the second condition.
  - 8 Select operator, and type a value, for example "F" (for female).
- 6 Click on the OK button to confirm your definitions and return to the main Transform screen. The new domain instruction appears in Main file, as shown in Figure 5-16.



**Figure 5-16** Creating a Domain instruction

- 7 Choose *Save* at the top of the Project screen to save the project.
- 8 Verify that the new column was created.
  - 1 Choose Data Management.
  - 2 Choose Explorer.
  - 3 Expand the list of tables and columns in the Repository tree. The new Domain column, (in this example FemalesUS), appears in the list. Select the column and choose View details to examine the new column properties and values, as shown in Figure 5-17.



**Figure 5-17** Validating the new domain column

## EXPRESSION

Using the *Expression* instruction, you can create calculated dynamic columns using one or two columns from a database. You can create *simple* expressions or *complex* expressions. For example, a *simple* expression concatenates strings by joining several columns. A *complex* expression uses mathematical operations such as logarithms.

### Parameters **Target table**

Define the target table where the expression column is created.

### **Target column**

Defines the name of the new column. Do not use the following characters: [ ] and " (also known as left square bracket, right square bracket, and double quotation mark).

### **Expression**

The expression used to create a column value. To create expressions you use operators, statements, and functions.

BIRT Analytics Loader supports the following operators described in Table 5-4.

**Table 5-4** Supported operators and statements

Name	Description	Example
+	Concatenate two text columns.	[Demo] . [Customer] . [Title] + [Demo] . [Customer] . [Surname]
+	Concatenate Numeric columns or strings.	STRING ( [CustomerID] ) +5+ [DaysSinceLastOrder] -1
+, -, *, /	Mathematical operators for adding, subtracting, multiplying, or dividing numeric columns.	STRING ( [Demo] . [Customer] . [Cust_ID] + [DEMO] . [Customer] . [Household_ID] )
GT[>], LT [<], GE[>=], LE[<=], EQ [=], NE[<>], AND, OR, NOT	Assign Boolean values to columns, based on the conditions.	[Demo] . [Customer] . [Edats] EQ "Adults">0 or [Demo] . [Customer] . [Edats] EQ "Joves">0
Constants	Assign a constant to a table column.	[Demo] . [Customer] . [Household_ID] =1
Boolean If statement	Return a value depending on a condition. The condition can be simple or more complex. The return value can be any type, for example yes or no.	If ( [DEMO] . [Order] . [OrderDate] > [Demo] . [Customer] . [DOB] , 1 , 0)

Table 5-5 presents the complete list of the functions. For more information about these functions, see *Using BIRT Analytics*.

**Table 5-5** Supported functions

<b>Name and syntax</b>	<b>Description</b>
ABS ([db].[table].[column])	Returns the absolute value of a numeric column, regardless of sign.
AGE ([db].[table].[column], date)	Returns the age of each value on a specified date.
ALLMONTHS ([db].[table].[column])	Returns the year and the month of each one of the values of a date column.
ALLQUARTERS ([db].[table].[column])	Returns the year and the quarter of each one of the values of a date column.
CEIL ([db].[table].[column])	Given a continuous numeric column, it returns the next integer value. For example, given the value 3.6, the CEIL function returns 4.
CHR ([db].[table].[column])	Given an integer numeric column, it returns the associated ASCII (American Standard Code for Information Interchange) value.
CODE ([db].[table].[column])	Returns the ASCII code for the first value of the text string. It only works for ASCII string columns, not for Unicode columns.
COUNT ([db].[table].[column])	Returns the number of records in a table.
DATE ([db].[table].[column], format)	Creates a date column converting input data to a date. You can use the following syntax: DATE(<string>   <unicode>, <date format:string>)
DATETIME ([db].[table].[column], format)	Creates a date column converting input data to a datetime. You can use the following syntax: DATETIME(<string>, <datetime format:string>) or DATETIME(<unicode>, <datetime format:string>)
DAY ([db].[table].[column])	Returns the part for the day in a date column.
DAYADD ([db].[table].[column], [db].[table].[column2])	Adds or removes the days required in a date column. The column returned is a date column.
DAYSTO ([db].[table].[column], date)	Returns the number of days between each value and a specified date.
DISTINCT ([db].[table].[column])	Returns the unique value or values in a column. For example, in a [Demo].[Customer].[Surname] column, it returns a single surname for each stored in the column.
EXP ([db].[table].[column])	Returns the exponential value of each of the values of a numeric column.
FLOOR ([db].[table].[column])	Returns the largest integer number less than or equal to the array.
HOUR ([db].[table].[column])	Returns the time of each date.
IF (A, B, C)	Returns a conditional function. A is the condition to be applied, B is the value to return if the condition is true, C is the value to return if the condition is false.
INSTR ([db].[table].[column], "string")	Returns an integer that specifies the initial position of the first appearance of a string in another one. First value is 0.
INT ([db].[table].[column], format)	Returns the integer number that forms part of the value in a column, or converts the column that contains the numbers into a numeric column. You can use the following syntax: INT(<string>   <unicode>, <thousands format:string>, <thousands grouping:integer   longint>)
ISNULL ([db].[table].[column])	Returns true for null values and false for not-null values.

**Table 5-5** Supported functions (continued)

<b>Name and syntax</b>	<b>Description</b>
ISNOTNULL ([db].[table].[column])	Returns true for not-null values and false for null values.
KURT ([db].[table].[column])	Returns the kurtosis of a column, which represents how values are distributed around the mean.
LCASE ([db].[table].[column])	Returns a lowercase text string, turning uppercase characters into lowercase.
LEFT ([db].[table].[column], n)	Returns the left part of a string of characters with the specified number of characters - where n is the quantity of characters. The length of the strings in the column, resulting from an expression using the LEFTfunction, will be defined by the parameter provided in the function. The only exception that can occur is when the parameter is provided through a variable whose value is defined by a certain column. In this case the column result for LEFT will have a length equal to the original column.
LEN ([db].[table].[column])	Returns the number of characters in a text column, returning a number for each value in the column.
LOG ([db].[table].[column])	Returns the natural logarithm (logarithm in base e) of the values of the column.
LOG10 ([db].[table].[column])	Returns the decimal logarithm (logarithm in base 10) of the values of the column.
LONGINT ([db].[table].[column], format)	Returns the long integer part of a column, or converts the column that contains a number into a numeric column. You can use the following syntax: LONGINT(<string>   <unicode>, <thousands format:string>, <thousands grouping:integer   longint>)
LTRIM ([db].[table].[column])	Returns a string that is a copy of a string with no initial spaces (deletes any initial spaces).
MAX ([db].[table].[column])	Returns the maximum value of a column. The column used must always be numeric (continuous or discrete).
MD5 ([db].[table].[column])	Returns an MD5 hash of a free text or a text field
MEAN ([db].[table].[column])	Returns the average value of a numeric column (continuous or discrete).
MEDIAN ([db].[table].[column])	Returns the value that divides the values of a column into two equal parts.
MID ([db].[table].[column], number1, number2)	Returns a string that contains a specified number of characters in a string. Two numbers are required: the first indicates the position where the string to be returned starts and the second indicates the number of characters to return (optional). If Number1 (the position indicator) is greater than the number of characters in the initial string, it returns a 0-length string. The length of the strings in the column, resulting from an expression using the MID function, will be defined by the parameter provided in the function. The only exception that can occur is when the parameter is provided through a variable whose value is defined by a certain column. In this case the column result for MID will have a length equal to the original column.
MIN ([db].[table].[column])	Returns the smallest value of the parameters of a numeric column (continuous or discrete).
MINUTE ([db].[table].[column])	Returns the minutes part of a date or time column.

**Table 5-5** Supported functions (continued)

<b>Name and syntax</b>	<b>Description</b>
MODE ([db].[table].[column])	Returns the value that is most repeated in the values of a column (the most frequent value).
MONTH ([db].[table].[column])	Returns the month part of a date column.
NOW ()	Returns the current date and time in the engine's default format.
POWER ([db].[table].[column], number)	Returns all the values of the column to the specified second parameter (number).
PROPER ([db].[table].[column])	Returns a new column in which all records begin with an uppercase letter and the rest are in lowercase.
PROPERSENTENCE ([db].[table].[column])	Returns a new column in which the first letter of each word in the description of the variables is in uppercase.
QUARTER ([db].[table].[column])	Returns the quarter of the year for each of the records in the date column.
RAND (n)	Returns a column based on random ( <i>n</i> ). For example, RAND (120) in the Customers table returns a column in which customers are randomly divided into 120 groups.
REAL ([db].[table].[column], format)	Returns the real value of a column. The source column must be a numeric column, a text string containing text, or a string representing a numeric column. You can use the following syntax: REAL(<string>   <unicode>, <decimal format:string>, <thousands format:string>, <thousands grouping:integer   longint>)
REGEXMATCH ([db].[table].[column], regular_expression[, "i"])	Returns 1 if the column containing a string value matches the regular expression and 0 otherwise. To perform a case-insensitive test, use the value "i" as the optional third argument. For information about the syntax of regular expressions and examples of their use, see "Using regular expression patterns to match and replace text strings."
REGEXREPLACE ([db].[table].[column], regular_expression, replacement[, "i"])	Returns a string containing the value of replacement in place of a matched regular expression in a column string value. To perform a case-insensitive test, use the value "i" as the optional fourth argument. For information about the syntax of regular expressions and examples of their use, see "Using regular expression patterns to match and replace text strings."
REPLACE ([db].[table].[column], pattern, replacement)	Replaces the pattern text with the replacement text. Both pattern and replacement can be columns or text
RIGHT ([db].[table].[column],n)	Returns the right part of a string of characters with the specified number of characters - where n is the quantity of characters.  The length of the strings in the column, resulting from an expression using the RIGHT function, will be defined by the parameter provided in the function. The only exception that can occur is when the parameter is provided through a variable whose value is defined by a certain column. In this case the column result for RIGHT will have a length equal to the original column.
ROUND ([db].[table].[column])	Rounds the input number to the nearest value containing no decimal positions. It creates a column of the same type as of the source column.
ROW ([db].[table].[column])	Returns a unique value, starting from 0, for each of the existing records in the column used to create the expression.
RTRIM ([db].[table].[column])	Returns a text string, an exact copy of the specified string without trailing spaces.
SECOND ([db].[table].[column])	Returns the seconds part of a date or time column.

**Table 5-5** Supported functions (continued)

<b>Name and syntax</b>	<b>Description</b>
SECONDSTO ([db].[table].[column], [db].[table].[column])	Returns a longint value that is the seconds elapsed between two columns of date or time format.
SECSTO ([db].[table].[column], [db].[table].[column])	Deprecated.
SGN ([db].[table].[column])	Returns the values grouped by -1 for negative values, 0 for invalid values, because not entered, for example, and +1 for positive values.
SHA1([db].[table].[column])	Returns an SHA1 hash of a free text or a text field.
SKEW ([db].[table].[column])	Returns the skewness value of the values of a column with respect to the mean value.
SQRT ([db].[table].[column])	Returns the square root of <i>n</i> , where <i>n</i> is the mean of the values of the column.
STDEV ([db].[table].[column])	Returns the standard deviation of the values of the column, which is the square root of the variance.
STRING ([db].[table].[column], format)	Returns a text string that represents any other data type, except Unicode. Use any of the following syntaxes: STRING(<integer>   <longint>   <real>, <decimal format:string>, <thousands format:string>, <thousands grouping:integer   longint>) STRING(<date> <date format:string>[, <date separator:string>]) STRING(<datetime>, <datetime format:string>) STRING(<datetime>, <datetime format:string>, <date separator:string>, <datetime separator:string>, <time separator:string>, <time decimal separator:string>) STRING(<time>, <time format:string>) STRING(<time>, <time format:string>, <time separator:string>, <time decimal separator:string>)
STRIP ([db].[table].[column])	Returns the entered column deleting any spaces.
SUM ([db].[table].[column])	Returns a new column with the cumulative sum of the values of the original column.
SUMSQ ([db].[table].[column])	Returns a new column with the cumulative sum of each of the values of the original column squared.
TIME ([db].[table].[column])	Returns the time of a string, date-and-time, or time column. You can use the following syntax: TIME(<string>   <unicode>, <time format:string>)
TODAY ()	Returns current date in engine's format.
TRIM ([db].[table].[column])	Returns a text string that contains the copy of the specified string with no spaces either at the beginning or the end of the string.
UCASE ([db].[table].[column])	Returns a new column with text in uppercase.

**Table 5-5** Supported functions (continued)

Name and syntax	Description
UNICODE([db].[table].[column], format)	Returns a unicode string that represents any other data type. You can use the following syntax: UNICODE(<integer>   <longint>   <real>, <decimal format:string>, <thousands format:string>, <thousands grouping:integer   longint>) UNICODE(<date>, <date format:string>[, <date separator:string>]) UNICODE(<datetime>, <datetime format:string>) UNICODE(<datetime>, <datetime format:string>, <date separator:string>, <datetime separator:string>, <time separator:string>, <time decimal separator:string>) UNICODE(<time>, <time format:string>) UNICODE(<time>, <time format:string>, <time separator:string>, <time decimal separator:string>)
WEEKDAY ([db].[table].[column])	Depending on the locale, returns the day of the week for each of the values. Default locale is en_US. For example, in the en_US locale: 1=Sunday. In es_ES locale: 1=Monday.
WEEKNUMBER ([db].[table].[column])	Returns the week number of a given date (as specified in ISO-8601).
YEAR ([db].[table].[column])	Returns the year of a date or date-and-time column.

### Using LONGINT with DATE, TIME, or DATETIME arguments

For DATE, TIME, or DATETIME arguments, you must use the LONGINT function together with the STRING function, as shown in the following examples:

```
Incorrect: LONGINT (DATE ("20130301"))
Correct: LONGINT (STRING (DATE ("20130301", "yyyymmdd"), "yyyymmdd"), "", 0)

Incorrect: LONGINT (TIME ("012345"))
Correct: LONGINT (STRING (TIME ("012345", "hhMMss"), "hhMMss"), "", 0)

Incorrect: LONGINT (DATETIME ("20120229 140600"))
Correct: LONGINT (REPLACE (STRING (DATETIME ("20120229 140600",
"yyyymmdd_hhMMss"), "yyyymmdd_hhMMss"), " ", ""), "", 0)
```

### Using INT with DATE, TIME, or DATETIME arguments

For DATE, TIME, or DATETIME arguments, you must use the INT function together with the STRING function, as shown in the following examples:

```
Incorrect: INT (DATE ("20130301"))
Correct: INT (STRING (DATE ("20130301", "yyyymmdd"), "yyyymmdd"), "", 0)

Incorrect: INT (TIME ("012345"))
Correct: INT (STRING (TIME ("012345", "hhMMss"), "hhMMss"), "", 0)
```

### Using format patterns for DATE, TIME, or DATETIME values

Format patterns are case sensitive and use the following components:

- yyyy – represents the year with 4 digits
- mm – represents the month with 2 digits
- dd – represents the day with 2 digits
- hh – represents the hour with 2 digits
- MM – represents the minute with 2 digits

ss – represents the second with 2 digits

xxx – represents the millisecond with 3 digits. By default, a decimal point (.) separates milliseconds from seconds, for example 12.345.

Underscore (\_) – represents any character

The following format patterns are supported when working with the DATE, TIME, or DATETIME functions:

**DATE**

ddmmyyyy

mmddyyyy

yyyymmdd

dd\_mm\_yyyy

mm\_dd\_yyyy

yyyy\_mm\_dd

**TIME**

hhMMss

hhMMssxxx

hh\_MM\_ss

hh\_MM\_ss\_xxx

**DATETIME**

yyyymmdd\_hhMMss

yyyymmdd\_hhMMssxxx

yyyymmdd\_hh\_MM\_ss

yyyymmdd\_hh\_MM\_ss\_xxx

yyyy\_mm\_dd\_hhMMss

yyyy\_mm\_dd\_hhMMssxxx

yyyy\_mm\_dd\_hh\_MM\_ss

yyyy\_mm\_dd\_hh\_MM\_ss\_xxx

ddmmyyyy\_hhMMss

ddmmyyyy\_hhMMssxxx

ddmmyyyy\_hh\_MM\_ss

ddmmyyyy\_hh\_MM\_ss\_xxx

dd\_mm\_yyyy\_hhMMss

dd\_mm\_yyyy\_hhMMssxxx

dd\_mm\_yyyy\_hh\_MM\_ss

dd\_mm\_yyyy\_hh\_MM\_ss\_xxx

mmddyyyy\_hhMMss

mmddyyyy\_hhMMssxxx

mmddyyyy\_hh\_MM\_ss

mmddyyyy\_hh\_MM\_ss\_xxx

mm\_dd\_yyyy\_hhMMss  
mm\_dd\_yyyy\_hhMMssxxx  
mm\_dd\_yyyy\_hh\_MM\_ss  
mm\_dd\_yyyy\_hh\_MM\_ss\_xxx

## Using regular expression patterns to match and replace text strings

The REGEXMATCH and REGEXREPLACE functions use arbitrarily complex regular expressions to find a string in a column value. A regular expression is a sequence of literal string values and pattern-matching symbols enclosed in quotation marks ("").

Table 5-6 lists the pattern-matching symbols and shows examples of their use.

**Table 5-6** Regular expression pattern-matching symbols

Symbol	Regular expression function	Examples
\	Enables or disables the regular expression function of the following character	"\" matches the   character "\" matches a horizontal tab character
	Separates alternative values	"gray   grey" matches "gray" and "grey"
()	Creates a group that defines operator precedence	"gr(a   e)y" matches "gray" and "grey"
?	Specifies zero or one of the preceding element	"colou?r" matches "color" and "colour"
*	Specifies zero or more of the preceding element	"ab*c" matches "ac", "abc", "abbc", and so on
+	Specifies one or more of the preceding element	"ab*c" matches "abc", "abbc", and so on, but does not match "ac"
{number}	Specifies an exact number of repetitions of the preceding element	"ab{2}c" matches "abbc" "(ab){2}c" matches "ababc"
		<i>(continues)</i>
{min,max}	Specifies the minimum and maximum repetitions of the preceding element	"ab{2,3}c" matches "abbc" and "abbbc" "(ab){2,3}c" matches "ababc" and "abababc"
{min,}	Specifies the minimum repetitions of the preceding element	"ab{2,}c" matches "abbc", "abbbc", "abbbbc", and so on
^	Requires the subsequent pattern to match the start of the string value	"^abc" matches "abc" in the string "abcdef" but not in the string "defabc"
\$	Requires the preceding pattern to match the end of the string value	"abc\$" matches "abc" in the string "defabc" but not in the string "abcdef"
.	Matches any single character	"a.c" matches "aac", "abc", "acc", and so on

**Table 5-6** Regular expression pattern-matching symbols (continued)

Symbol	Regular expression function	Examples
[character expression]	Specifies an expression that matches a single character. The expression contains one or more characters or ranges of characters. The expression interprets other pattern-matching symbols as simple characters.	"gr[ae]y" matches "gray" and "grey" "[abc]" matches "a", "b", and "c" "[a.c]" matches "a", ".", and "c" "[a-z]" matches any lowercase letter from "a" to "z" "[abcx-z]" matches "a", "b", "c", "x", "y", and "z" "[a-zA-Z]" matches any lowercase or uppercase letter
[^character expression]	Specifies an expression that matches any single character not in the expression.	"[^a-zA-Z]" matches any character that is not a lowercase or uppercase letter, for example "1" or "@"

Table 5-7 lists and describes escape-sequence patterns that match specific characters or classes of characters.

**Table 5-7** Regular expression escape-sequence patterns

Pattern	Represents	Examples
\b	Backspace character (0x08)	
\e	Escape character (0x1B)	
\n	Newline character(0x0A)	
\r	Return character (0x0D)	
\t	Horizontal tab character (0x09)	
\v	Vertical tab character (0x0B)	
\d	Any decimal digit character: 0-9	
\D	Any character not a decimal digit	
\h	Any hexadecimal digit character: 0-9, a-f, and A-F	
\H	Any character not a hexadecimal digit	
\s	Any white-space character: space, horizontal tab, vertical tab, return, newline, formfeed (0x0C), next line (0x85)	"^\s+" matches white-space characters at the start of a string
\S	Any character not a white-space character	"\S\$" matches a single non-white-space character at the end of a string
\w	Any alphanumeric, underscore (_), or multibyte character	
\W	Any character not an alphanumeric, underscore (_), or multibyte character, such as control characters (0x01 through 0x1F) and punctuation characters	
\digit	A back reference to an element enclosed in parentheses	"(abc)def(\1)" matches "abcdefabc"

The following examples show more complex usage of regular expressions as arguments to the REGEXMATCH and REGEXREPLACE functions:

```
REGEXMATCH([Demo].[Customer].[column], "H(ä|ae?)ndel")
```

Returns 1 for strings that match the words "Handel", "Händel", and "Haendel".

```
REGEXMATCH([Demo].[Customer].[column], "^[\s]+|[\s]+$")
```

Matches excess whitespace (space or tab) at the beginning or end of a line.

```
REGEXMATCH([Demo].[Customer].[column],
"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,3}$")
```

Matches an e-mail address for which the user name part includes punctuation characters.

```
REGEXMATCH([Demo].[Customer].[column], "^\\w+@[a-zA-Z_]+?\\. [a-zA-Z]{2,3}$")
```

Uses the \\w sequence to match an e-mail address for which the user name part does not include punctuation characters.

```
REGEXMATCH(STRING([Demo].[Customer].[Income]),
"^[+-]?(\d+\.?\d*|\.\d+)([eE][+-]?\d+)?$")
```

Matches any number.

```
REGEXREPLACE(UNICODE([Demo].[Customer].[Income]),
"^[+-]?\d+\.?\d*$", "\1")
```

Replaces a real number by its integer part.

### How to create an expression

This example creates a column that shows the creation of a column of the year a customer first made their purchase.

- 1 In the *Transform* tab, choose *New*. The list of transformation instructions appears.
- 2 In the list of transformation instructions, select *Expression*. *Expression* appears, as shown in Figure 5-18.

**Figure 5-18** Creating an expression

- 3 In the *Table* fields, select a target database and a table.
- 4 In the *Column* field, type the name of the new column.
- 5 In the *Expression* field, type the expression.
- 6 In the *Description* field, type the description of the column.
- 7 Click "OK" to confirm your choices and return to the main *Transform* screen. The *Expression* instruction appears in *Main* file.
- 8 Choose "Save" at the top of the *Loader Project* window to save the project.

- 9 After running the project, verify that the new column was created in Explorer in the *Data Management* tab.

## INUMERIC

Using numeric ranges, you can create a dynamic column that groups values into numeric ranges. A numeric range must be based on a numeric column. Numeric ranges are useful for simplifying data. For example:

- There is a column in the database containing *profit per order*. You can group these figures into five intervals such as very high, high, medium, low, and very low.
- Another very common case is grouping individuals into *age ranges*. These ranges tend to be very different depending on the type of analysis you want to do, and so being able to configure the ranges manually gives you the flexibility you need.

### Parameters

#### Source column

Defines a database, table, and column for a source for the numeric ranges. The column must be of numeric type. Column names must not contain the following characters: [ ] and " (also known as left square bracket, right square bracket, and double quotation mark).

#### Target column

Defines the name of the target column. The column is added to the source table. Do not use the following characters: [ ] and " (also known as left square bracket, right square bracket, and double quotation mark).

#### Bands

Defines the ranges in the following format:

```
[Lower range number] | [Upper range number] | [Range Name]
```

Each range must be on a separate line. The lower and upper numbers define the numbers to include in the range. The first element, or lower range number (From) is ">=" (greater than or equal to) while the second element, or upper range number (To) is "<" INUMERIC (less than).

For example, to define three age groups in households, you use the following definition:

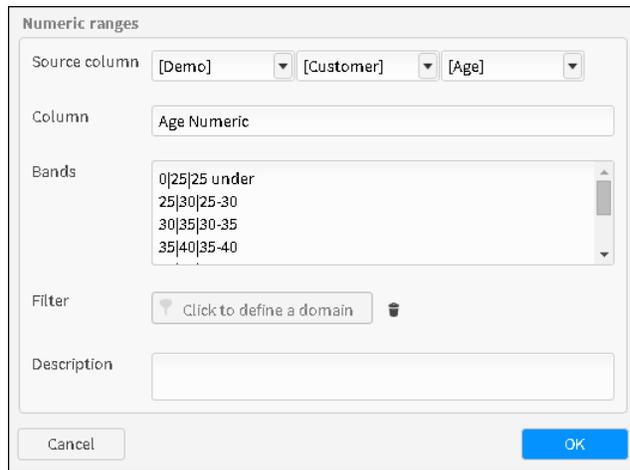
```
1 | 30 | Young  
30 | 55 | Middle-age  
55 | 105 | Old
```

#### Filter

Defines or uses an existing domain to filter the data set in the table.

#### How to create numeric ranges

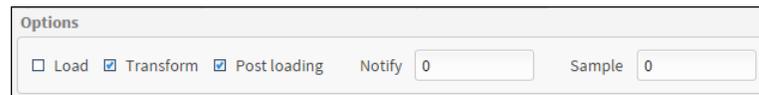
- 1 In the *Transform* tab, choose "New". The list of transformation instructions appears.
- 2 In the list of transformation instructions, select INUMERIC. the *Numeric ranges* screen appears, as shown in Figure 5-19.



**Figure 5-19** Creating numeric ranges

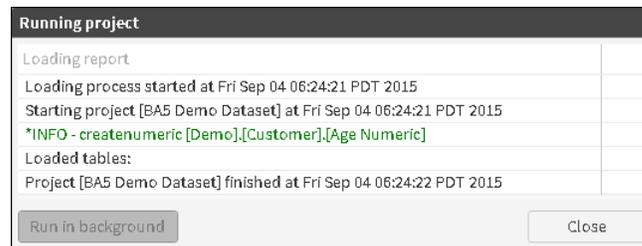
- 3 In the *Source column* fields, select a database, a table, and a column that will be used as a source for the numeric range column. The column must be numeric.
- 4 In the *Column* field, type the name of the new numeric range column.
- 5 In the *Bands* field, define the ranges and their names. Double-click in the box to open the text editor and type the text.
- 6 Click “OK” to confirm your choices and go back to the main *Transform* tab screen. The INUMERIC instruction appears in *Main file* panel.
- 7 Save the project.
- 8 Run the instruction without loading data.

- 1 In the *Main file* panel, select each instruction. Then select the (-) icon, to make each one inactive.
  - 2 Remove the selection from the Load option in the Data tab, as shown in Figure 5-20.



**Figure 5-20** Removing the Load option

- 3 Click on the Run tool at the top of your project page. Then select “Yes’ in the confirmation screen that opens. As the project runs, messages describing each stage of the project appear in Running project, as shown in Figure 5-21.



**Figure 5-21** Running the INumeric instruction

- 9 Verify that the new column was created in Explorer in Data Management.

## PARAMETRIC

Parametric columns consist of query-based values. These queries are formulated with values or columns belonging to any table in the database as long as the tables are properly joined. The resulting dynamic column is created in the table indicated during the process, so the results are very different depending on the resolution marked by the selected table.

For example, in the Demo database, you can determine sales of a particular product. For the Customers table, the query returns the customers who bought one or more quantities of the product, while for the Products table the query returns how many X products have been sold.

The number of products sold is probably greater than the number of customers who have bought them, because one customer may buy more than one unit of a product, but one unit of a product cannot be sold to more than one customer.

A parametric column is not displayed in the data explorer, since the same record may meet the condition of several values in the parametric column.

### Parameters **Target table**

Defines a database, and table.

### **Target column**

Defines the name of the target column. Do not use the following characters: [ ] and " (also known as left square bracket, right square bracket, and double quotation mark).

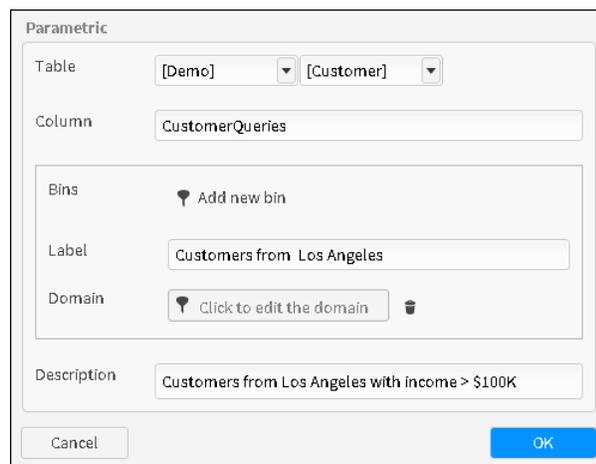
### **Bins**

A set of data values that fall in a given interval. Define a bin name and a query that fills the bin with data. To define the query, use Domain.

### **How to create a parametric column**

This example creates a parametric column, which groups the customers into two bins, customers from Los Angeles, and customers with an income greater than \$100K.

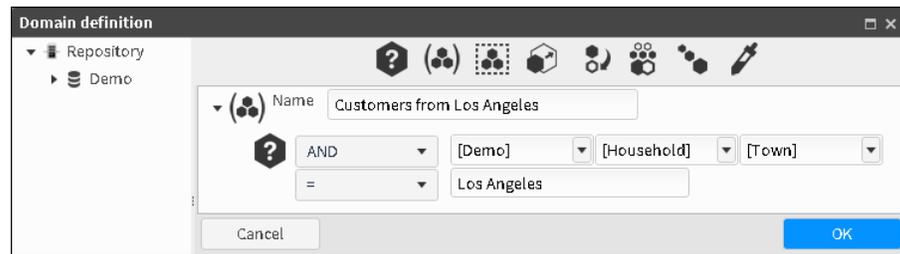
- 1 Choose "New" in the *Transform* tab. The list of transformation instructions appears.
- 2 In the list of transformation instructions, choose PARAMETRIC to open the Parametric screen shown in Figure 5-22.



**Figure 5-22** Creating a parametric column

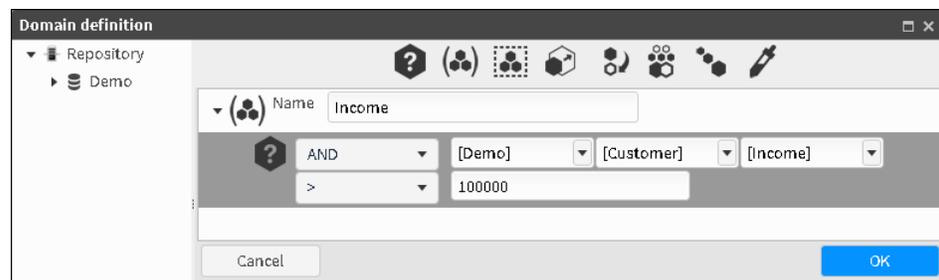
- 3 In the *Table* fields, select the target database and table.
- 4 In the *Column* field, type the name of the new column.
- 5 Define a new bin.
  - 1 Click on "Add new bin" in the *Bins* field to expand it to include the *Label* and *Domain* fields.

- 2 In the *Label* field, type the name of the new bin, for example *Customers from CA*.
- 3 Click on the *Domain* field to open the *Domain definition* screen, as shown in Figure 5-23.



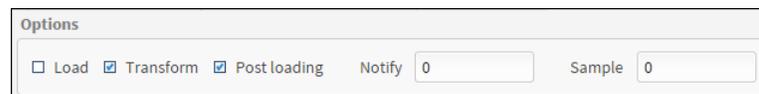
**Figure 5-23** Creating the Customers Los Angeles domain

- 4 In the *Domain definition* screen, define the query and choose “OK”.
- 5 Back in the *Parametric* screen, choose “Add new bin” again, to define another bin.
- 6 Define a *label* for the domain, and select the *Domain* field to define the query. The *Domain definition* screen appears again, as shown in Figure 5-24.



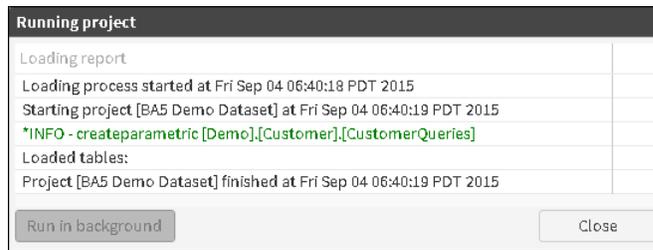
**Figure 5-24** Creating the CreditLimit domain

- 7 Choose “OK” in the *Domain definition* screen to return to the *Parametric* screen.
  - 6 Now choose “OK” in the *Parametric* screen to go back to the main *Transform tab* screen, The *Parametric* instruction appears in the *Main file* panel
  - 7 Save the project.
  - 8 Run the instruction without loading data.
- 1 In the *Main file* panel, select each instruction. Then choose (-) to make them inactive, as shown in Figure 5-25.
  - 2 In *Data—Options*, remove the selection for *Load* box, as shown in Figure 5-25.



**Figure 5-25** Removing the Load option

- 3 Choose *Run*. As the project runs, messages describing each stage of the project appear in *Running project*, as shown in Figure 5-26.



**Figure 5-26** Running a Parametric instruction

- 9 Verify that the new column was created in Explorer in Data Management.

## QUANTILE

A quantile range dynamic column is created from a numeric column. QUANTILE creates groups that contain the same number of values. These groups are created by sorting the values in ascending order so that the first range contains the smallest values and the last range contains the largest values. For example, you can identify customers that generate the greatest profit.

### Parameters **Source column**

Defines a numeric column.

### **Target column**

Defines the name of the new target column. Do not use the following characters: [ ] and " (also known as left square bracket, right square bracket, and double quotation mark).

### **Number of parts**

Defines the parts the data must be grouped into.

### **Filter**

Defines a domain, if the data requires filtering.

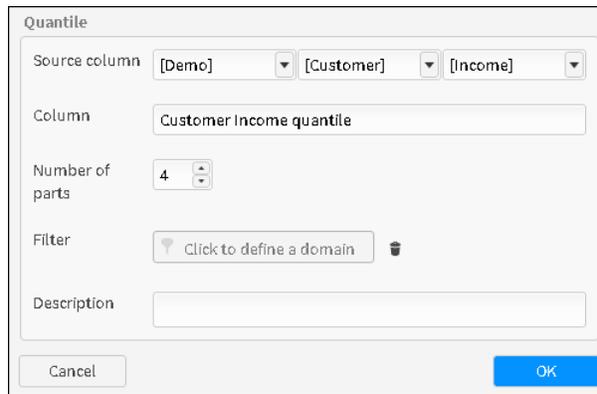
### **How to create a quantile range column**

The following example describes how to create a quantile column that creates four groups that span the range of income values found in a set of customer records. Table 5-8 lists the resulting quantile ranges and the customer records in each range.

**Table 5-8** Quantile ranges

#	Income range	Number of records
1	479-14081	56232
2	14082-19627	56232
3	19628-36390	56232
4	36391-113334	56232

- 1 In the *Transform* tab, choose "New". The list of transformation instructions appears.
- 2 In the list of transformation instructions, choose QUANTILE. The *Quantile* screen appears, as shown in Figure 5-27.

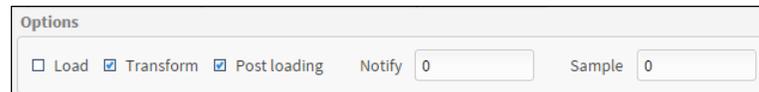


**Figure 5-27** Creating a quantile column

- 3 In the *Source column* fields, select a database, a table, and a numeric column.
- 4 In the *Column* field, type the name of the new column.
- 5 In the *Number of parts* field, use the arrows to provide quantiles, groups, or ranges that the column will contain.
- 6 Click on the *Filter* field to define a domain, if you need to filter the data.
- 7 Click “OK” in the *Quantile* screen to confirm your choices and go back to the main *Transform tab* screen. The *Quantile* instruction appears in *Main* file.
- 8 Save the project.
- 9 Run the instruction without loading data.

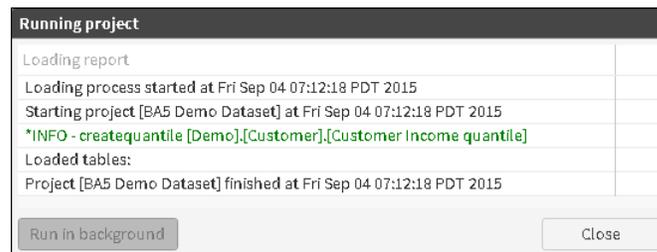


- 1 In *Main* file, select each instruction. Then choose (-) to make them inactive, as shown in Figure 5-28.
- 2 Remove the selection for the *Load* box in the *Data* tab, as shown in Figure 5-28.



**Figure 5-28** Removing the Load selection

- 3 Run the project. Messages describing each stage of the project appear in the *Running project* screen, as shown in Figure 5-29.



**Figure 5-29** Running a Quantile instruction

## RANKING

Supports creating a new column with the rank for each subset of records relating to the same key. Ranking enables you to rank a table based on a specific column. For each subset of records associated with the same key in a table, a sequence number is assigned according to the relative value of a given column. Ranking is used, for example, when trying to identify the first products purchased by each customer, or the most recent product purchased.

Ranking creates a column with an integer value, which ranks records in the same table.

**Parameters**

**Key to rank**

Defines a column that represents the key to be ranked. The column must be an index key in a table.

**Column to use**

Defines the column to use when creating the ranking numbers.

**Column name**

Defines the name of the new ranking column. Do not use the following characters: [ ] and " (also known as left square bracket, right square bracket, and double quotation mark).

**Sorting direction**

Defines the sorting direction, ASC or DESC.

**Force sequence**

Defines the ranking sequence. In cases where two or more records have exactly the same value, selecting Force sequence ensures that each record is allocated a different rank. If you do not check Force sequence, records with the same value have the same rank.

**Filter**

Defines a domain in case you want to filter the data.

**How to create a ranking column**

- 1 In the *Transform* tab, choose "New". The list of transformation instructions appears.
- 2 In the list of transformation instructions, choose RANKING. The *Ranking* screen appears, as shown in Figure 5-30.

The screenshot shows a 'Ranking' dialog box with the following configuration:

- Key to rank:** Three dropdown menus showing 'Demo', '[Customer]', and '[Cust ID]'.
- Field to use:** Three dropdown menus showing 'Demo', '[Order]', and '[Order Date]'.
- Name:** Text box containing 'CustomerPlacedOrders'.
- Direction:** Dropdown menu set to 'ASC'.
- Force sequence:** Unchecked checkbox.
- Filter:** Button labeled 'Click to define a domain' and a trash icon.
- Description:** Text box containing 'Rank customers, depending on the date they placed orders.'

**Figure 5-30** Creating a ranking column

- 3 In the *Key to rank* fields, select a database, a table, and a column you use as a key for ranking.
- 4 In the *Field to use* field, select a database, a table, and a column you use to determine the ranking values.
- 5 In the *Name* field, type the name of the new ranking column.
- 6 In the *Direction* field, select the table-sorting direction from the drop-down list.
- 7 Activate the *Force sequence* box.
- 8 Click on the *Filter* field to define a domain, if you want to filter the data.
- 9 Type a short description of the column in the *Description* field.

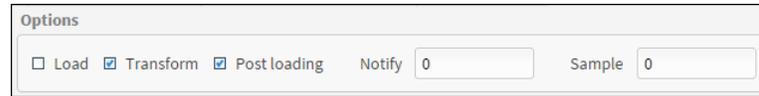
10 Click “OK” to confirm. The Ranking instruction appears in *Main file* panel

11 Save the project.

12 Run the instruction without loading data.



- 1 In Main file, select each instruction. Then choose the (-) icon to make it inactive.
- 2 Remove the selection from the Load box in the Data tab, as shown in Figure 5-31.



**Figure 5-31** Removing the Load option

- 3 Choose Run. The project runs and a confirmation message appears.

## UNPIVOT

Create a new table by unpivoting columns from a source table.

Allows creating a new table by unpivoting columns from a source table. **Table** is the name for the new table. **Discrete column** is the column from the source table that is the primary key. It must be discrete. **Columns to unpivot** is a list of numeric columns (integer or longinteger) to use for unpivoting. **Filter** is an optional filter.

For example, you could generate a table with all the dates included in a date interval. In a source table with a primary key (PK) and with columns **startdate** and **enddate**, the user can generate a target table with all PK and all the dates in the interval between **startdate** and **enddate**.

**Parameters** **Target table**  
Defines a database, and table.

**Target column**  
Defines the name of the target, discrete column. Do not use the following characters: [ ] and " (also known as left square bracket, right square bracket, and double quotation mark).

**Columns to unpivot**  
A List of numeric columns (integer or long integer) to use for unpivoting.

**Filter**  
Optional. Defines a domain in case you want to filter the data.

---

## Managing data objects

The instructions in this group work with data objects such as: databases, tables, and columns.

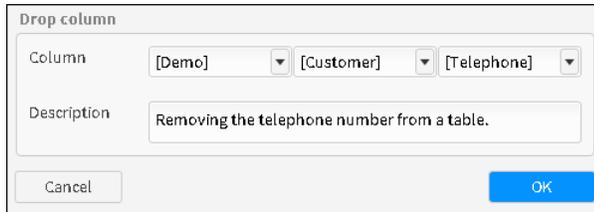
### DROPAUTOLINKS

Deletes all transient links in the repository.

### DROPCOLUMN

Removes a column from a table.

**Parameter** **Column name**  
Define the name of the column you want to drop from a table, as shown in Figure 5-32. Do not use the following characters: [ ] and " (also known as left square bracket, right square bracket, and double quotation mark).



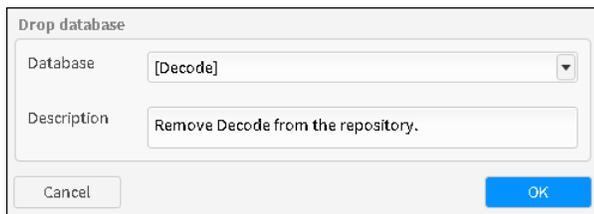
**Figure 5-32** Removing a column from a table

## DROPDATABASE

Removes a database from the repository.

**Parameter Database name**

Defines the name of the database you want to remove from the repository, as shown in Figure 5-33.



**Figure 5-33** Removing a database from the repository

## DROPPERIVEDCOLUMNS

Deletes all derived columns from a table.

**Parameter Table name**

Defines the name of the table you want to remove derived columns.

## DROPLINK

Removes a link between two tables. Use Droplink to delete existing relations between columns in different tables.

**Note:** *If the links have not been created or have been created incorrectly, it is not possible to work with various levels of resolution in the analysis.*

**Parameters From column name**

Defines the name of the column from which the join is initiated.

**To column name**

Defines the name of the column to which the link is created. Figure 5-34 shows Drop link.



**Figure 5-34** Removing a link between tables

## DROPTABLE

Removes a table from the repository.

**Parameter** **Table name**

Defines the name of the table you want to remove from a database, as shown in Figure 5-35.



**Figure 5-35** Removing a table from a database

## INDEX

Creates an index for a column in a table. This function is useful when a project is appending data to a table coming from distinct sources or files and Loader is instructed not to index columns. After all data is loaded, calling this function indexes the columns. Otherwise, Loader indexes incrementally, which generally takes longer.

**Parameter** **Column name**

Defines the database, table, and column that will be used to create an index.

## UNINDEX

Removes column indexing.

**Parameter** **Column name**

Defines the database, table, and column that will have its index removed.

## LINK

Creates relations between columns in different tables. Links can also be created using the Links option in Data Management. If the links have not been created or have been created incorrectly, it is not possible to work with various levels of resolution in the analysis.

***Note:** In the case of links with a 1 to 1 relationship, the first column becomes the Parent table and the second column becomes the Child table.*

Creates a link between two tables.

**Parameters** **From column name**

Defines the name of the column from which the join is initiated.

**To column name**

Defines the name of the column to which the link is created.

**How to create a link between two tables**

The example creates a link between two tables, Customers and SalesData on CustomerNumber.

- 1 In the *Transform* tab, choose "New". The list of transformation instructions appears.
- 2 In the list of transformation instructions, choose LINK. The *Link* screen appears, as shown in Figure 5-36.

**Figure 5-36** Creating a link between two tables

- 3 In the *From* field, define a database, a table, and a column the link initiates.
  - 4 In the *To* field, define a database, a table, and a column to which the link points.
  - 5 Click “OK” to return to the *Transform tab* screen. The new LINK instruction appears in the *Main file* panel.
  - 6 Save the project.
  - 7 Run the instruction without loading data.
- ➔
- 1 In the *Main file* panel, select each instruction. Then choose the (-) icon to make it inactive.
  - 2 Remove the selection from the *Load* box in the *Data* tab, as shown in Figure 5-37.

**Figure 5-37** Removing the Load selection

- 3 Run the project. Messages describing each stage of the project appear in the *Running project* screen.

## RANGEDLINK

Creates a ranged link between two tables.

Allows a link between two tables to be created using a pair of columns on the primary table which denotes a range, to a column in the secondary table. Lower value is the name of the column in Table 2 to be linked (lower range). Upper value is the name of the column in Table 2 to be linked (upper range). Many is the name of the column in Table 1 to be linked.

**Parameters** **Lower value**  
Name of the column in Table 2 to be linked – lower range.

**Upper value**  
Name of the column in Table 2 to be linked – upper range.

**Many**  
Name of the column in Table 1 to be linked

## MAKEPERMANENT

The columns created by transformation functions are dynamic by nature and contain the formulas used for their definition, instead of real data. MAKEPERMANENT makes a dynamic column permanent, which means the column will be filled with data.

**Parameter Column name**

Defines the column name you want to make permanent, as shown in Figure 5-38. Provide the full column definition in the following format:

[Database] [Table] [Column]

Note that the dynamic columns do not appear in the column's drop-down list. You must type the column name, surrounded by square brackets.

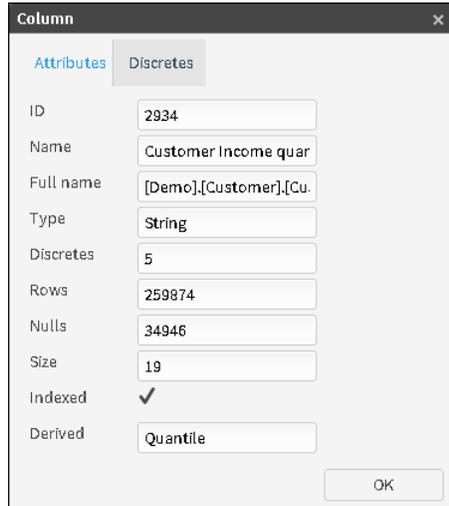


The screenshot shows a dialog box with three dropdown menus at the top, each containing a column name in square brackets: [Demo], [Customer], and [Customer Inco]. Below these is a text field labeled 'Description'. At the bottom, there are 'Cancel' and 'OK' buttons.

**Figure 5-38** Making a column permanent

You can verify that the column has been converted to permanent in Explorer. Select the column properties and confirm its *Derived* property is not set to *Not Derived*.

For example, the [Customer Income quantile] column is a dynamic column, created using the QUANTILE transformation instruction. The column's *Derived* property is *Quantile* before it is made permanent, as shown in Figure 5-39.



The screenshot shows a 'Column' dialog box with two tabs: 'Attributes' and 'Discretizes'. The 'Attributes' tab is active. It contains several fields: ID (2934), Name (Customer Income quar), Full name ([Demo],[Customer],[Cu.]), Type (String), Discretizes (5), Rows (259874), Nulls (34946), Size (19), Indexed (checked), and Derived (Quantile). An 'OK' button is at the bottom right.

**Figure 5-39** Dynamic column status

After the transformation the column status is changed to *Not derived*, as shown in Figure 5-40.

**Figure 5-40** Permanent column status

## RENAME

Renames a database object, such as a database, a table, or a column.

**Parameters** **Old name**

Defines a current name of a database, a table, or a column in the following format:

[Database] . [Table] . [Column]

**New name**

Defines a new name of a database, a table, or a column in the following format:

[Database] . [Table] . [Column]

Rename is shown in Figure 5-41.

**Figure 5-41** Renaming a column

## Miscellaneous instructions

The instructions in this group have different purposes.

### EXPORT

Exports a domain to a table or a file.

**Parameters** **Export target and Target name**

Defines the object to where the data is exported. The export target can be:

- *Analytical database and table*  
Defines new or existing database and table names.

- *CSV file*

Defines a CSV file name. The user sets the desired delimiter by selecting it in the dropdown list of available delimiters. They are: apos, at, colon, coma, equal, minus, pipe, plus, quote, semicolon, sharp, tab, tilde)

The user can choose where the file is to be stored by setting a specific export folder, together with the file name (*c:\Actuate\exportfile.txt* for example).By default, the file is stored in:

```
C:\Program Files\BIRAnalytics\data\FastDB
```

- *Fixed-length file*

Defines a file name. The columns have fixed length and the file is saved by default in:

```
C:\Program Files\BIRAnalytics\data\FastDB
```

The user can also choose where to store fixed-length files in the same way as for CSV files.

**Query resolution table**

Defines the table you are exporting data from.

**Query columns**

Defines the query columns. Each column must be defined on a separate line, using the full name in the following format:

```
[Database] . [Table] . [Column]
```

To export the whole table, do not define columns.

**Filter**

Defines a domain you are using to filter the data. Filter is a required parameter.

**How to export to a database**

You have to choose the database where you want the new table to be created, the name of the new table, and the columns to be created in the destination table.

- 1 In the *Transform* tab, choose "New". The list of transformation instructions appears.
- 2 In the list of instructions, choose EXPORT. The *Export* screen appears, as shown in Figure 5-42.

**Figure 5-42** Exporting data to a table

- 3 In the *Export target* options, select Table. As you can see in Figure 5-42, several fields are not activated for this option.
- 4 In the *Export target* fields, type the new database and table names either by typing them into the fields or selecting them from the drop-down lists. There are also two checkboxes that allow you to select (or not select) the use of headers (*using headers*) or the use of short column names (*short column names*).
- 5 In *Query* options, *Resolution table* fields, select the table from which you want to export data.
- 6 In the *Query* options, *Enter columns* field, type the name of the columns, using the required format. It is also possible to drag columns from the *Repository structure* panel and drop them in this field. Leave the field empty if you want to export the whole table. You can double-click in the field to open a text editor and type the column list.
- 7 Click on the *Click to define a domain* field to define a domain that would filter the data. Defining a domain is required.
- 8 In the *Description* field, type a short description of the instruction's purpose.
- 9 In the *Order by* panel, choose the column that you want to sort with and the direction of the sort: ascending or descending.
- 10 Click "OK" to confirm your choices and go back to the main *Transform* screen.
- 11 Save the project.

12 Run the Export instruction.

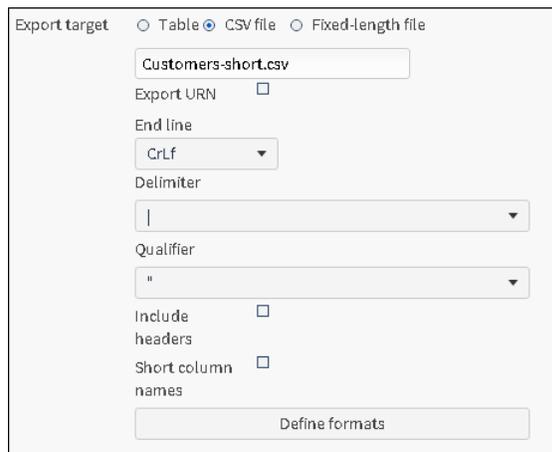
13 To verify the export, go to Explorer in the *Data* tab and refresh the database. The new table with the chosen column data should appear.

#### How to export data to a text file

1 In the *Transform* tab, choose “New”. The list of transformation instructions appears.

2 In the list of transformation instructions, choose EXPORT. The *Export* screen appears again. This time choose “CSV file” as your Export target and type the name of your CSV file in the *Target* field at the top.

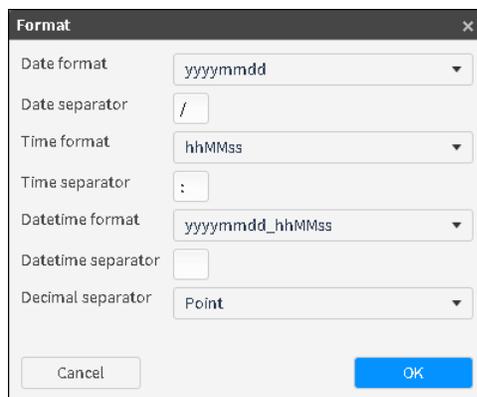
3 Now the previously disabled CSV configuration fields are accessible and need to be filled in as in figure Figure 5-43. Here we chose “CrLf” for the *End line* entry, the “pipe” symbol (|) for the *Delimiter* entry and a quotation mark (") for the *Qualifier* entry. You can also choose to *include headers* or to use *short column names* by selecting the corresponding checkboxes.



The screenshot shows the 'Export target' configuration dialog. At the top, there are three radio buttons: 'Table', 'CSV file' (which is selected), and 'Fixed-length file'. Below this, there is a text input field containing 'Customers-short.csv'. Underneath are two checkboxes: 'Export URN' (unchecked) and 'Include headers' (unchecked). The 'End line' is set to 'CrLf' in a dropdown menu. The 'Delimiter' is set to '|' in a dropdown menu. The 'Qualifier' is set to '"' in a dropdown menu. Below these are two more checkboxes: 'Short column names' (unchecked) and 'Define formats' (a button).

Figure 5-43 Exporting data to configured CSV file

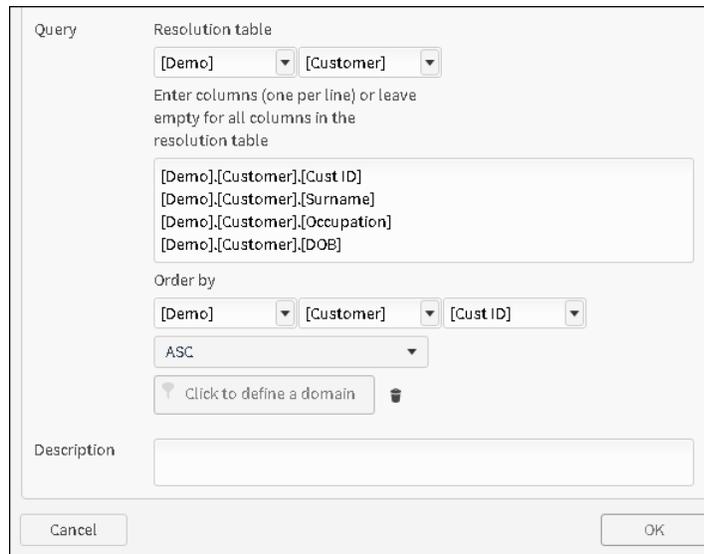
4 Click on the *Define formats* field to open the *Format* screen and define your desired formats as shown in Figure 5-44. Here you can set your *local* date, time and datetime formats if needed.



The screenshot shows the 'Format' dialog box. It has several fields: 'Date format' set to 'yyyyMMdd', 'Date separator' set to '/', 'Time format' set to 'hhMMss', 'Time separator' set to ':', 'Datetime format' set to 'yyyyMMdd\_hhMMss', 'Datetime separator' (empty), and 'Decimal separator' set to 'Point'. At the bottom, there are 'Cancel' and 'OK' buttons.

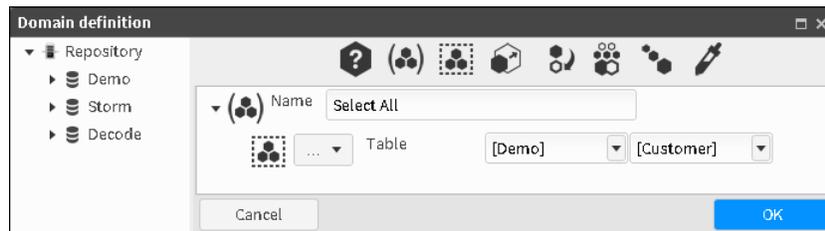
Figure 5-44 Defining your local date, time and datetime formats

5 Fill in the rest of the fields, as shown in Figure 5-45.



**Figure 5-45** Entering the Export information

- 6 In the *Resolution table* fields, select the table from which you want to export data.
- 7 In the *Enter columns* field, type the name of the columns, using the required format. Leave it empty if you want to export the whole table. You can also drag and drop columns here from the *Repository structure* panel.
- 8 Click in the *Click to define a domain* field to define a domain that would filter the data, as shown in Figure 5-46. Defining a domain is required.



**Figure 5-46** Selecting domain information

Choose OK when you finished defining the domain.

- 9 In the *Description* field, type a short description of the instruction's purpose.
- 10 Click "OK" to confirm your definition choices.
- 11 Save the project.
- 12 Run the Export instruction. The exported file is now saved on the machine where BIRT Analytics Loader is installed. The file is located in:

C:\Program Files\BIRTAnalytics\data\FastDB

As you can see in the sample of the file contents, the chosen file formats have been respected:

```
13 |Lilly|OW|19420709
21 |Aucote|PR|19620113
53 |Wathers|SE|19560606
58 |Lampit|MW|19610818
```

#### How to export to a fixed-length file

- 1 In the *Transform* tab, choose "New". Then choose EXPORT in the list of transformation instructions that appears.

- 2 The same *Export* screen that we worked with in the previous *Export* procedures appears again. This time choose “Fixed length file” as your Export target and type the name of this target file in the *Target* field at the top.
- 3 Now the *End line* field is accessible and needs to be filled in as in Figure 5-47. Here we chose “CrLf” for the *End line* entry and “Customer-short.txt” as the name of the Target file.

The screenshot shows the 'Export target' dialog box. At the top, there are three radio buttons: 'Table', 'CSV file', and 'Fixed-length file', with 'Fixed-length file' selected. Below this, there is a text input field containing 'Customers-short.txt'. Underneath is a checkbox for 'Export URN' which is unchecked. The 'End line' section has a dropdown menu currently showing 'CrLf'. Below that is a 'Delimiter' dropdown menu showing '|'. The 'Qualifier' dropdown menu shows '"'. There are two more checkboxes: 'Include headers' (unchecked) and 'Short column names' (unchecked). At the bottom of the dialog is a 'Define formats' button.

**Figure 5-47** Exporting data to a fixed-length file

- 4 Fill in the rest of the fields as shown in Figure 5-48.

The screenshot shows the 'Resolution table' dialog box. At the top, there are two dropdown menus for 'Resolution table' containing '[Demo]' and '[Customer]'. Below this is a text area with the instruction 'Enter columns (one per line) or leave empty for all columns in the resolution table'. The text area contains four lines of column specifications: '[Demo],[Customer],[Cust ID]', '[Demo],[Customer],[Surname]', '[Demo],[Customer],[Occupation]', and '[Demo],[Customer],[DOB]'. Below the text area is an 'Order by' section with three dropdown menus containing '[Demo]', '[Customer]', and '[Cust ID]', and a sorting dropdown set to 'ASC'. There is a button labeled 'Click to define a domain' with a trash icon. At the bottom is a 'Description' text field. The dialog has 'Cancel' and 'OK' buttons at the bottom.

**Figure 5-48** Entering the Export information

- 5 In the *Resolution table* fields, select the table from which you want to export data.
- 6 In the *Enter columns* field, type the name of the columns, using the required format. Leave it empty if you want to export the whole table.
- 7 Click in the *Click to define a domain* field to define a domain that would filter the data. Defining a domain is required.
- 8 In the *Description field*, type a short description of the instruction’s purpose.
- 9 Click “OK” to confirm your definition choices.
- 10 Save the project.

**11** Run the Export instruction. The exported file is saved on the machine where BIRT Analytics Loader is installed. The file is located in:

C:\Program Files\BIRTAnalytics\data\FastDB

**12** The file format uses fixed length for saving the data column.

## IFEXISTS

Conditionally executes an INCLUDE statement, depending on the value of an expression. The expression evaluates the existence of a database object, such as a database, a table, and a column. Depending on the result, it executes the *true* or *false* INCLUDE statement.

### Parameters Object

A database object such as database, table, or column.

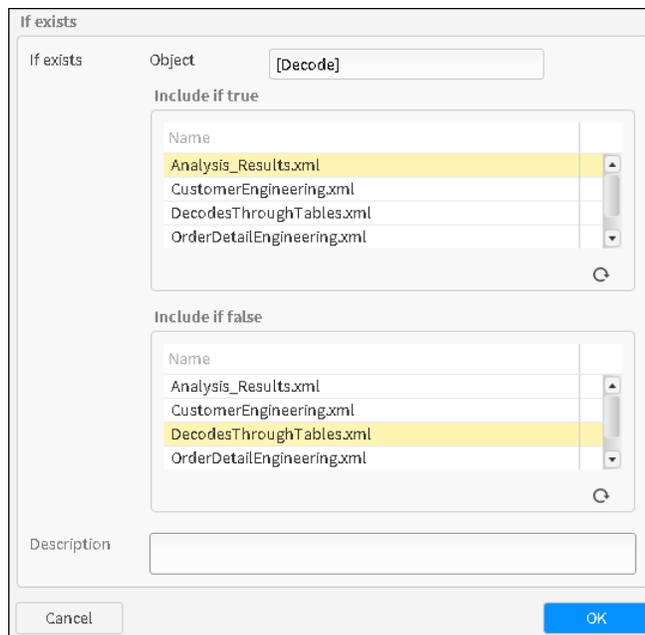
### Include if true

The file name of a script file. If the database object exists, the instruction includes this script file in the load process.

### Include if false

The file name of a script file. If the database object does not exist, the instruction includes this script file in the load process.

The example shown in Figure 5-49 checks if the [Decode] database exists. The IFEXISTS instruction executes the *Analysis\_Results.xml* script if the table exists, or *DecodesThroughTables.xml*, if the table does not exist.



**Figure 5-49** Creating the IFEXISTS instruction

## IFLINKED

Evaluates the existence of a link between two tables and executes an INCLUDE statement, depending on the result. This instruction enables you to dynamically execute scripts, depending on a condition.

## Parameters

### From link

Defines the name of the column from which the join is initiated. The column must be defined using the full name in the following format:

[Database] . [Table] . [Column]

### To link

Defines the name of the column to which the link is created. The column must be defined using the full name in the following format:

[Database] . [Table] . [Column]

### Include if true

The file name of a script file. If the link exists, the instruction includes this script file in the load process.

### Include if false

The file name of a script file. If the link does not exist, the instruction includes this script file in the load process.

## How to use IFLINKED

- 1 In the *Transform* tab, choose “New”. In the list of transformation instructions, choose IFLINKED. The *If linked* screen appears.
- 2 In *From*, define a database, a table, and a column from which the link initiates, as shown in Figure 5-50.

The screenshot shows the 'If linked' dialog box. It has a title bar 'If linked'. Below the title bar, there are two rows of dropdown menus. The first row is labeled 'From' and contains three dropdowns: '[Demo]', '[Customer]', and '[Household ID]'. The second row is labeled 'To' and contains three dropdowns: '[Demo]', '[Household]', and '[Household ID]'. Below these are two sections: 'Include if true' and 'Include if false'. Each section has a 'Name' label and a list of script files. In the 'Include if true' section, the list contains: 'Analysis\_Results.xml', 'CustomerEngineering.xml', 'DecodesThroughTables.xml', and 'Household\_Engineering.xml'. The 'Household\_Engineering.xml' file is highlighted in yellow. In the 'Include if false' section, the list contains: 'Analysis\_Results.xml', 'CustomerEngineering.xml', 'DecodesThroughTables.xml', and 'Household\_Engineering.xml'. The 'CustomerEngineering.xml' file is highlighted in yellow. Below these sections is a 'Description' label and an empty text box. At the bottom left is a 'Cancel' button, and at the bottom right is an 'OK' button.

**Figure 5-50** Creating an IFLINKED instruction

- 3 In *To*, define a database, a table, and a column, to which the link points.
- 4 In *Include if true*, select the script file to be executed if the link exists. Choose *Refresh* if you do not see your script in the list.
- 5 In *Include if false*, select the script file to be executed if the link does not exist. Choose *Refresh* if you do not see your script in the list.
- 6 In *If linked*, choose OK. The *Iflinked* instruction appears in Main file.

# INCLUDE

Includes a script file or creates an empty include entry.

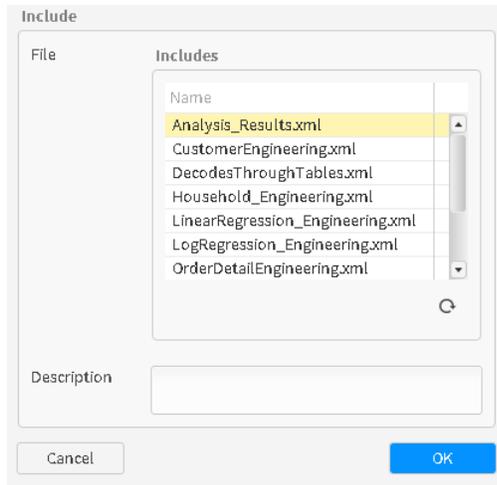
**Parameter** **File name**

The name of the script file you want to include in the main script.

**How to include a file**

The example includes a script file into the main script.

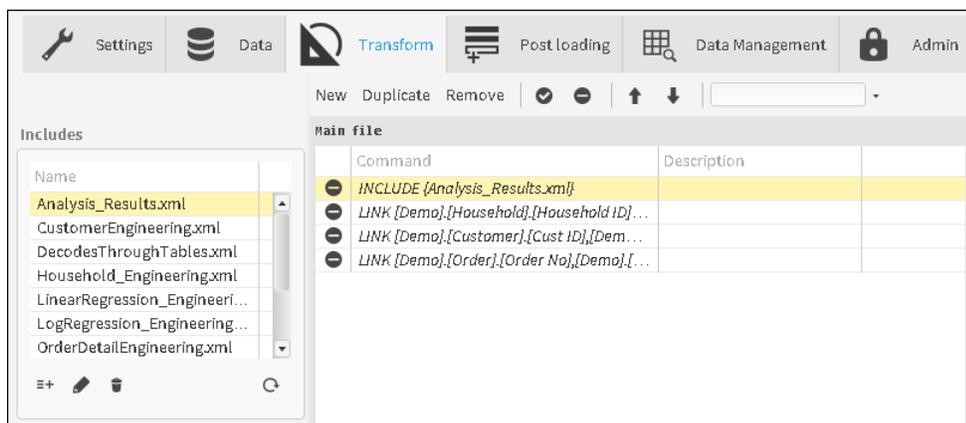
- 1 In the *Transform* tab, choose “New”. The list of transformation instructions appears.
- 2 In the list of transformation instructions, choose “Include.” The *Include* screen appears, as shown in Figure 5-51.



**Figure 5-51** Including a script file



- 3 Click on the *Refresh* icon to make the latest script files appear.
- 4 In the *Name* field, select the file you want to include.
- 5 In the *Description* field, write a short note explaining the instruction.
- 6 Click “OK” to confirm your definition and go back to the main *Transform* screen. The new instruction appears in the *Main file* panel, as shown in Figure 5-52.



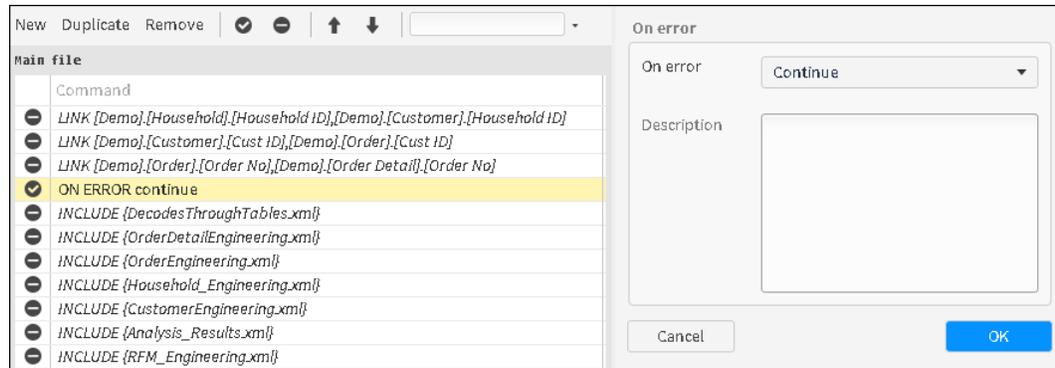
**Figure 5-52** Reviewing included scripts

## ON ERROR

Allows the user to choose between continuing or breaking the load process when an error occurs.

By default, the loading process stops whenever an error is encountered. The On Error instructions let you choose whether you want the transformation process to stop in the case of error or to continue running. If you choose to continue, the transformation process ignores any columns that present errors during the process, leaving them to be fixed and processed later if desired.

Use the “up” and “down” arrows at the top of the *Transform* tab to position your instructions wherever you want them to be in your rows of instructions. Figure 5-53 shows an on error instruction to continue.



**Figure 5-53** Managing exceptions using the Transform tab

## PRINT

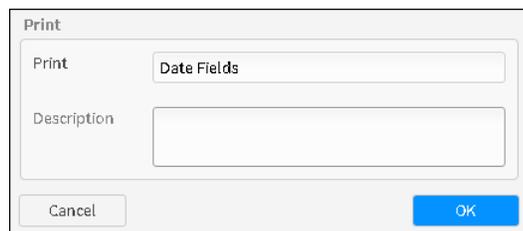
Writes comments in a project log file.

**Parameter** **Text**

The text to include in the log file.

For example, a comment, like the one shown in Figure 5-54, appears in the log files as:

```
09/07/2015 01:37:02 - *INFO - Date Fields
```



**Figure 5-54** Creating a comment

## VAR

Defines variables. Variables can be used to dynamically control script execution. You can store different types of values in variables.

**Parameters** **Variable name**

The name of the variable. Use descriptive names.

**Variable value**

The value of the variable. The values can be of any type.

To reference a variable use the following syntax:

%VAR%

### How to use a variable

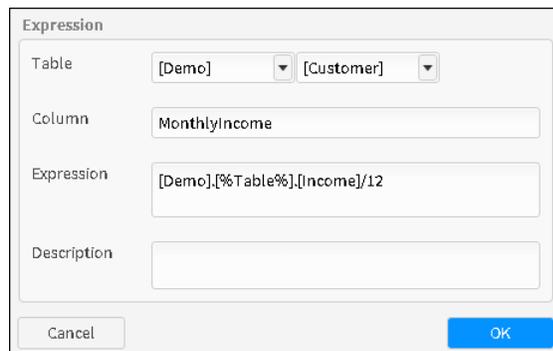
- 1 In the *Transform* tab, choose “New”. Then choose VAR in the list of transformation instructions that appears. The *Variable* screen appears.
- 2 In the *Name* field, type the variable name, as shown in Figure 5-55. In the *Value* field, type the variable value. In this example, the variable contains the name of a table.



The screenshot shows a dialog box titled "Variable". It has two input fields: "Name" with the text "Table" and "Value" with the text "Customer". Below these is a larger "Description" field. At the bottom, there are "Cancel" and "OK" buttons.

**Figure 5-55** Creating a variable

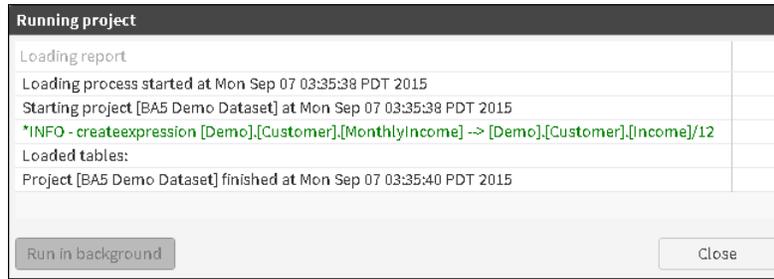
- 3 Click “OK” to confirm your definition and go back to the main *Transform* screen. The new VARIABLE instruction appears in the *Main file* panel.
- 4 In the *Transform* tab, choose “New” and this time choose EXPRESSION in the transformation instructions list that appears. This opens the *Expression* screen.
- 5 In the first *Table* field, select a database from the drop-down list. Then type the following expression [%Table%] in the second *Table* field, as shown in Figure 5-56:



The screenshot shows a dialog box titled "Expression". It has a "Table" field with two dropdown menus, the first showing "Demo" and the second showing "[Customer]". Below is a "Column" field with the text "MonthlyIncome". The "Expression" field contains the text "[Demo].[%Table%].[Income]/12". There is a "Description" field below. At the bottom, there are "Cancel" and "OK" buttons.

**Figure 5-56** Using a variable

- 6 In the *Column* field, type the name of the new column.
- 7 In the *Expression* field, type the expression for calculating the value of the new column. Use the variable reference where needed.  
[Demo] . [%Table%] . [Income]
- 8 Click “OK” to confirm you definition and go back to the main *Transform* screen.
- 9 Save your project.
- 10 Run the instruction. The log, as shown in Figure 5-57, shows that the TABLE variable is replaced by its value, Customer.



**Figure 5-57** Run log



# 6

## Administering BIRT Analytics Loader

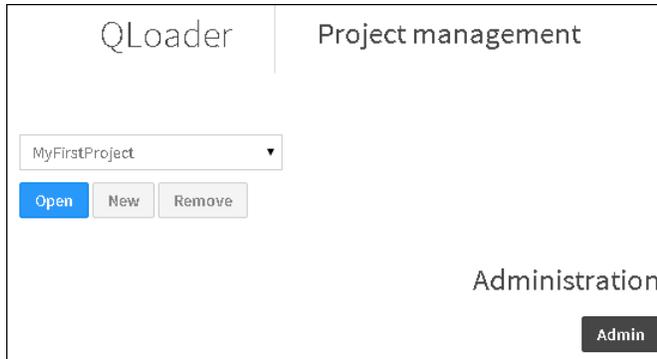
This chapter contains:

- About BIRT Analytics Loader administration
- Managing users
- Scheduling load processes
- Using External call
- Organization of storage in Loader
- Copying a load project to a new location
- Managing the repository

---

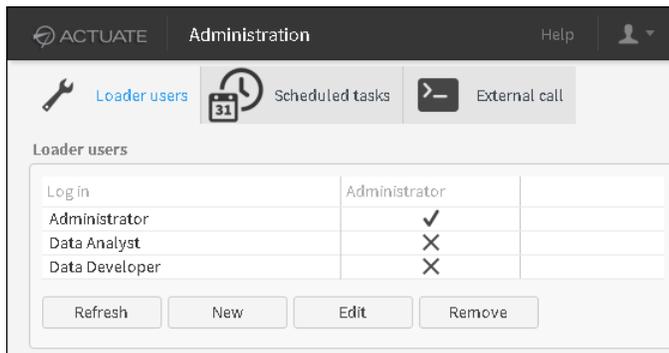
## About BIRT Analytics Loader administration

To access BIRT Analytics Loader Administration, you must have an administration account. Administration is located on the front page of the BIRT Analytics Loader, as shown in Figure 6-1, and appears only if the logged-in user is an administrator.



**Figure 6-1** Starting Administration

The Administration page of BIRT Analytics Loader is shown in Figure 6-2.



**Figure 6-2** Administering BIRT Analytics Loader

The main activities are presented as tabs:

- **Loader users**  
Manages the BIRT Analytics Loader users.
- **Scheduled tasks**  
Create and manage BIRT Analytics Loader scheduled tasks.
- **External call**

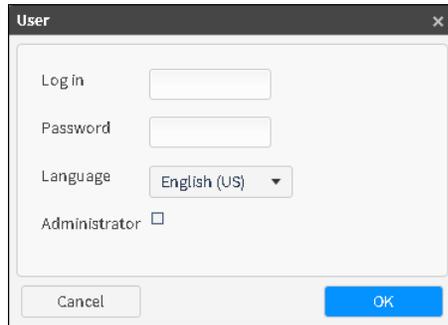
Returns a credential security file for a selected project. Using the file enables you to run a project remotely.

---

## Managing users

The *Loader users* tab is enables user access to BIRT Analytics Loader. There are four options:

- **Refresh**  
Refreshes the user content.
- **New**  
Create a new BIRT Analytics Loader user, as shown in Figure 6-3.



**Figure 6-3** Creating a user

For each user, you must define the following:

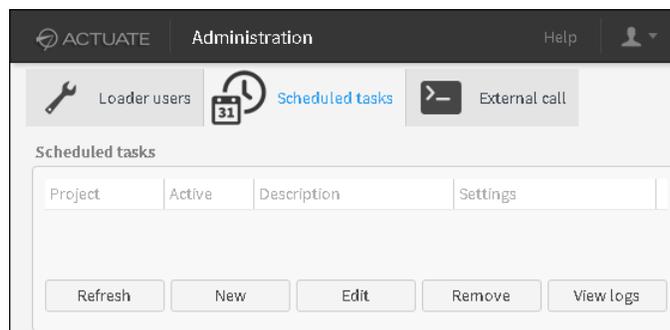
- **Login**  
User name.
- **Password**  
Password for the user.
- **Language**  
The language in which you want the user interface to appear. Available languages are: Catalan, English (US), French, German, Japanese, Russian, Simplified Chinese, Spanish and Turkish.
- **Administrator**  
Indicate if the user is an administrator.
- **Edit**  
Edit an existing BIRT Analytics Loader user.
- **Remove**  
Remove a BIRT Analytics Loader user.

---

## Scheduling load processes

Scheduling enables you to run the load processes overnight, or at times when the system is idle when BIRT Analytics users are not it.

The *Scheduled tasks* tab enables you to create and manage scheduled tasks, as shown in Figure 6-4.



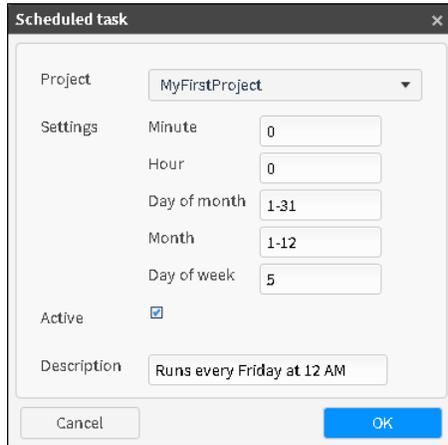
**Figure 6-4** Scheduling tasks

The following activities are available:

- **Refresh**

Refresh scheduled tasks.

- **New**  
Create new tasks. When creating a new task, you must provide the following information, as shown in Figure 6-5:



**Figure 6-5** Creating a scheduled task

- **Project**  
Select a project name.
- **Settings**  
Time settings, such as minute, hour, day of month, month, and day of week. Time settings use the following syntax:
  - **Minute**  
Minutes within the hour (0-59)
  - **Hour**  
The hour of the day (0-23)
  - **Day of month**  
The day of the month (1-31)
  - **Month**  
The month (1-12)
  - **Day of week**  
The day of the week (0-7), where both 0 and 7 are Sunday

To specify multiple values for one column, the following operators are available, as listed in Table 6-1, in the order of precedence.

**Table 6-1** Scheduling operators

Operators	Description	Examples
*	Match all values of a field.	Using an asterisk in Month indicates every month.
M-N	Define a range.	1-5 in Day of week means from Monday to Friday.
*/X	Specify multiple values based on a start position and a period.	*/15 in Minute selects 0, 15, 30, 45 minutes. 3-59/15 in Minute indicates the third minute of the hour and every 15 minutes thereafter.

**Table 6-1** Scheduling operators

Operators	Description	Examples
A,B,...,Z	Commas are used to separate items of a list.	0,30 for minutes, or 1,3,5 for days of the week.

- Active  
The task does not run if Active is not selected.
- Description  
A short note describing the scheduled task.
- Edit  
Edit tasks.
- Remove  
Remove tasks.
- View logs  
View execution logs, as shown in Figure 6-6.



**Figure 6-6** Viewing scheduled task log

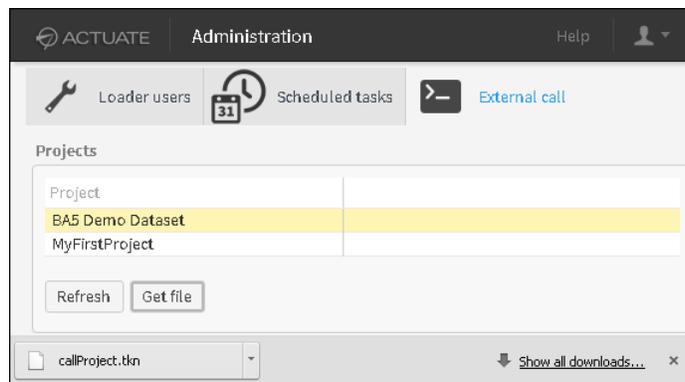
## Stopping the Tomcat service

BIRT Analytics installs with Tomcat Apache. To prevent end users from logging in to BIRT Analytics, stop the web server on which BIRT Analytics runs, using either a command-line editor or Windows Services.

---

## Using External call

*External call* is used to display a credential security file for a project, as shown in Figure 6-7. The credential file enables you to run a project remotely, from a command prompt, or a batch file. Typically, projects contain load and transformation instructions. You can control what instructions are executed from Data and Transform options in Projects management.



**Figure 6-7** Getting a credential security file

The credential file contains encrypted information about the current date and time, the project name, and security information associated with the system where the product is installed. This information includes security keys that the BIRT Analytics Loader installer generates and saves in this folder:

```
C:\Program Files\BIRTAalytics\data\FastDB\Loader
```

Public-key encryption uses two different mathematically related keys, known as a key pair, to encrypt and decrypt data. Information encrypted using one key can only be decrypted by using the other member of the key pair. The names of the keys are:

- `qloader.pbl`  
Public key, associated with the system where the product is installed
- `qloader.prv`  
Private key, used to decrypt the information encrypted with the public key

These files are only valid on the server where they are created. If these files are deleted, you can regenerate them by reloading the BIRT Analytics Loader web application to the Tomcat web server. Every time you reinstall the product, you must regenerate the load processes again.

The following activities are available:

- **Select project**  
A drop-down list of the projects is available.
- **Get file**  
Generates a credential token for the selected project. This token launches load processes remotely, or from batch files. By default, the project file name is `callProject.tkn`. You can change the name by choosing a name specific for a loading project.

## Running projects using qLauncher

The qLauncher is a Java-based utility that runs projects from batch files or a command prompt. The utility launches a load process remotely. The file, `qLauncher.jar`, is in the Tools folder of the installation files. `qLauncher.jar` is built using Java 1.8, and requires Java 1.8 in order to run.

The qLauncher command syntax is:

```
java -jar qlauncher.jar --url=[BIRT Analytics Loader Web Service URL]
--file=[Credential project file name] --out=[log file name]
```

### Parameters

**--url**  
The BIRT Analytics Loader web service URL. This is a required parameter.

**--file**  
The name of the project credential file. This is a required parameter.

**--out**  
The name of the file to save the execution log. This is an optional parameter.

Example:

```
java -jar qlauncher.jar --url="http://localhost:8110/qloader/run"
--file="callProject.tkn" --out="output.txt"
```

The `qLauncher.jar` file must be placed in the same folder as the project credential file. To obtain help for the command syntax, open a command prompt, and type the following:

```
java -jar qlauncher.jar
```

When you run a project, the utility changes the public key with the private key to log in to the BIRT Analytics Loader engine. To automate the data loading, include the command in a batch file.

### How to run a project using qLauncher

This procedure runs a project remotely.

- 1 Generate a project credential file.
  - 1 Log in to Admin in BIRT Analytics Loader.
  - 2 In External call, select a project from the project list.
  - 3 Choose Get file. This opens the following screen used for opening and saving your project as a file names *callProject.tkn* file.

Depending on the web browser, the file is saved in the configured download folder.

- 2 Create a folder for the remote project, run on your computer, for example:

```
C:\Load\qLauncher
```

- 3 Copy qLauncher.jar from the Tools folder of your installation files to the loading folder.

- 4 Copy callProject.tkn to the loading folder.

- 5 Open a command prompt CMD.

- 6 In CMD, navigate to the project folder. For example, type:

```
cd C:\Load\qLauncher
```

- 7 In CMD, type:

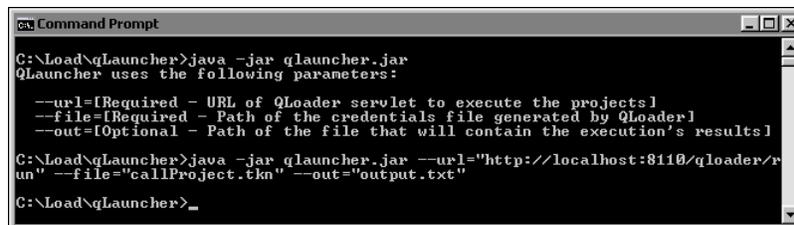
```
java -jar qLauncher.jar
```

The help for the command syntax appears, as shown in Figure 6-8. This step is optional.

- 8 In CMD, type the command to run the project. In this example the command looks like this:

```
java -jar qlauncher.jar --url="http://localhost:8110/qloader/run"  
--file="callProject.tkn" --out="output.txt"
```

The project run finishes and the prompt appears, as shown in Figure 6-8.



```
cs. Command Prompt  
C:\Load\qLauncher>java -jar qlauncher.jar  
qLauncher uses the following parameters:  
--url=[Required - URL of Qloader servlet to execute the projects]  
--file=[Required - Path of the credentials file generated by Qloader]  
--out=[Optional - Path of the file that will contain the execution's results]  
C:\Load\qLauncher>java -jar qlauncher.jar --url="http://localhost:8110/qloader/run"  
--file="callProject.tkn" --out="output.txt"  
C:\Load\qLauncher>
```

Figure 6-8 Running qLauncher.jar

- 9 qLauncher saves the log file, output.txt, in the current folder. Listing 6-1 shows the log outputs for the executed transactions.

#### Listing 6-1 output.txt

```
- Loading process started at Mon Aug 27 14:12:30 PDT 2015  
08/27/2015 14:12:31 - Starting project [MyFirstProject] at Mon Aug 27  
14:12:31 PDT 2015  
08/27/2015 14:12:31 - *INFO - Table: [SalesData].[Customers]  
08/27/2015 14:12:34 - *INFO - Table: [SalesData].[Customers] - 122  
08/27/2015 14:12:34 - *INFO - Table: [SalesData].[CarInventory]  
08/27/2015 14:12:35 - *INFO - Table: [SalesData].[CarInventory] - 110
```

```

08/27/2015 14:12:35 - *INFO - Table: [SalesData].[DailyPurchases]
08/27/2015 14:12:36 - *INFO - Table: [SalesData].[DailyPurchases] - 35
08/27/2015 14:12:36 - *INFO - Table: [SalesData].[MonthlySales]
08/27/2015 14:12:36 - *INFO - Table: [SalesData].[MonthlySales] - 109
08/27/2015 14:12:36 - *INFO - Table: [SalesData].[SaleOrders]
08/27/2015 14:12:38 - *INFO - Table: [SalesData].[SaleOrders] - 326
08/27/2015 14:12:46 - *INFO - Table: [SalesData].[Employees]
08/27/2015 14:12:47 - *INFO - Table: [SalesData].[Employees] - 14
08/27/2015 14:12:48 - *INFO - Table: [SalesData].[Clients]
08/27/2015 14:12:49 - *INFO - Table: [SalesData].[Clients] - 122
08/27/2015 14:12:49 - Loaded tables:
    Customers, CarInventory, DailyPurchases, MonthlySales, SaleOrders,
    Employees, Clients
08/27/2015 14:12:49 - Project [MyFirstProject] finished at Mon Aug 27
14:12:49 PDT 2015

```

---

## Organization of storage in Loader

BIRT Analytics Loader configuration information is stored in `dubnium.db`, an internal SQLite database which requires no intervention by BA Analytics users.

The scripts, logs, and repository generated by BIRT Analytics Loader are stored in the FastDB installation folder. By default, these folders are:

- Repository

`C:\Program Files\BIRTAnalytics\data\FastDB\databases\`

- Log Files

`C:\Program Files\BIRTAnalytics\log\`

- Load Project

`C:\Program Files\BIRTAnalytics\data\FastDB\loading-projects\`

Contains load projects. Every project is stored in a different folder. The names of these folders is a sequential number, such as 1, 2, 3, and so on.

A project folder contains the following sub-folders and files:

- Data

A folder where you upload and store source files with data.

- Transformations

A folder where the transform files are stored. These files are generated in XML format.

- Definitions

A folder where the definitions of files and column information are stored. These files are generated in XML format.

- Descriptor.txt

An XML file that contains the project name and its creation date.

- Control.xml

An XML file that contains the declaration of files to load.

- Postloading.xml

An XML file that contains the post loading parameters.

---

## Copying a load project to a new location

This section explains how to copy the files of a load project between two separate environments. Use this feature to move the project folder location, copy projects between machines, and move projects from development to production.

The project folders are named using sequential numbers like 1, 2, 3, and so on. You must identify your project by its content.

### How to copy a load project

- 1 Create a new empty load project using BIRT Analytics Loader. The configuration.xml, as shown in Listing 6-2, defines the path where the load project is stored.

#### Listing 6-2 Configuration.xml

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration>
<configuration>
  <projectspath>C:\Program Files\BIRTAnalytics\data\FastDB\loading-
  projects</projectspath>
  <descriptor>descriptor.txt</descriptor>
  <urlclient>fastdbloader/build/client.jsp</urlclient>
  <urladmin>admin/build/client.jsp</urladmin>
  <title>BIRT Analytics QLoader</title>
  <transformationslist>transformations.xml</transformationslist>
  <ignoresslerrors>1</ignoresslerrors>
  <!-- uploading files. Unit is Kb -->
  <maxmemsize>1024</maxmemsize>
  <!-- uploading files. Unit is Kb. -1 or 0 mean no limit -->
  <maxfilesize>-1</maxfilesize>
  <connectionstringsfile>connectionstrings.xml</connectionstringsfile>
  <!-- electron WS location and namespace -->
  <electronurl>http://localhost:8110/electronws/services</electronurl>
  <electronqname>http://services.ws.electron.ba.actuate.com</
  electronqname>
  <electronns>com.actuate.ba.electron.ws.services</electronns>
  <!-- information -->
  <baversion>5</baversion>
</configuration>
```

- 2 It would be wise to close the recently created project now.
- 3 Replace the files in the new load project with your old project files, (except for the descriptor.txt file).
- 4 Open the load project from BIRT Analytics Loader and verify that it works properly.

---

## Managing the repository

The FastDB database installation creates a default data repository in this path:

```
C:\Program Files\BIRTAnalytics\data\FastDB\databases
```

You have the option to work in the default repository or create a new one. To back up a repository, copy the content of the \databases folder to the backup location. In case of a large repository, compress the repository folder to minimize the required space. Larger repositories require more time to compress and copy. Consider this factor when you plan a backup because your system is not accessible during the operation.

### How to back up a repository

- 1 Stop the BIRT Analytics - FastDB service.
- 2 Make a copy of the current repository content in the following location:  
C:\Program Files\BIRTAnalytics\data\FastDB\databases
- 3 Start the BIRT Analytics - FastDB service.

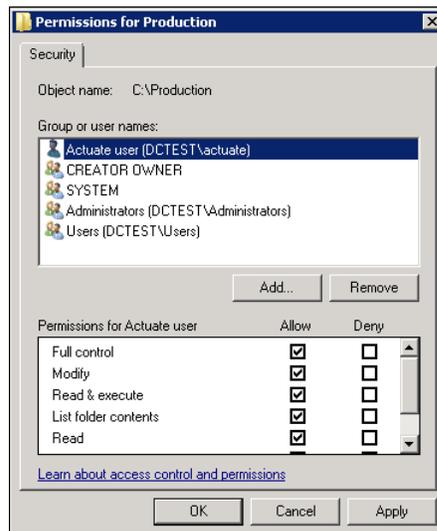
### Creating a new repository

To create a new FastDB database you must specify the database location in the configuration files. At installation, the default database contains the Demo database, and is located at:

C:\Program Files\BIRTAnalytics\data\FastDB\databases

### How to create a new repository

- 1 Stop the BIRT Analytics - FastDB service.
  - 1 In Windows→Start, choose Services
  - 2 In Services, select the BIRT Analytics - FastDB service, and choose Stop the service.
- 2 In Windows Explorer, create a new folder where you want to store the new FastDB database. For example:  
C:\Production\SalesData
- 3 Set privileges for the user running the FastDB server to this folder.
  - 1 In Windows Explorer, select the folder, and choose Properties from the right-click menu.
  - 2 In Properties, choose Security.
  - 3 In Security, choose Edit.
  - 4 In Group or user names, select the user running the FastDB server, as shown in Figure 6-9. If the user is not in the list, choose Add and select the user to be added.



**Figure 6-9** Setting folder privileges

- 5 In Permissions for <user>, in Full control, select Allow. Choose Apply. Choose OK.  
In Security, choose OK.  
In Properties, choose OK.

- 4 Open Windows Explorer, and navigate to:

```
C:\Program Files\BIRAnalytics\data\FastDB\
```

- 5 Open the engine\_configuration.ini file in Notepad. The file looks like the one in Listing 6-3.

**Listing 6-3** engine\_configuration.ini

---

```
[engine]
license = C:\Program Files\BIRAnalytics\data\FastDB\dubnium.lic
repository = C:\Program Files\BIRAnalytics\data\FastDB\databases
exchange_path = C:\Program Files\BIRAnalytics\data\FastDB\exchange

[log]
path = C:\Program Files\BIRAnalytics\log\engine.log
verbosity = 6
dbfibrillator_path = C:\Program Files\BIRAnalytics\log\dbfibrillator.log

[server]
admin_file_path = C:\Program Files\BIRAnalytics\data\FastDB\admin.sqlite
port = 8105
threads = 2

[memory]
maxmemorysystem = 3199
maxmemorythread = 2399
```

- 6 Modify the following line:

```
repository = C:\Program Files\BIRAnalytics\data\FastDB\databases
```

to:

```
repository = C:\Production\SalesData
```

- 7 Save and close the file.

- 8 In Services, select the BIRT Analytics - FastDB service and choose Start the service.

## Moving a repository

BIRT Analytics supports changing the location of a data repository by copying the repository to a new location. The FastDB server must be shut down before copying the repository. Before restarting the FastDB server, you must change the FastDB engine configuration file to specify the new location.

### How to copy a data repository to a new location

- 1 Stop the BIRT Analytics - FastDB service.

- 2 Open Windows Explorer, and navigate to:

```
C:\Program Files\BIRAnalytics\data\FastDB\databases
```

- 3 Open the engine\_configuration.ini file in Notepad.

- 4 Modify the following line:

```
repository = C:\Program Files\BIRAnalytics\data\FastDB\databases
```

for example, to:

```
repository = C:\Production\SalesData
```

- 5 Set privileges for the user running the FastDB server to this folder, in this case:

```
C:\Production\SalesData
```

To set privileges, follow the procedure in step 3 in “How to create a new repository,” earlier in this chapter.

- 6 Save and close the file.
- 7 Copy all files and folders from the old repository, which by default is:  
C:\Program Files\BIRTAnalytics\data\FastDB\databases  
to the newly defined repository:  
C:\Production\SalesData\
  - 8 Start the BIRT Analytics - FastDB service.
  - 9 Synchronize and assign permissions using BIRT Analytics Administration, as described in “Synchronizing data” in Chapter 2, “Working with projects.”
  - 10 Open BIRT Analytics and verify that it works properly.

## Restoring a repository

You can change the repository content without changing the location. For example, you have a backup of a BIRT Analytics repository, and you need to restore the data to the same repository location. In this case, it is not necessary to change the configuration file, you only change the content of the repository folder.

### How to change the content of a repository

This procedure restores the content of the FinancialData repository to the current Demo repository. The FinancialData backup is in the C:\Financial folder.

- 1 Stop the BIRT Analytics - FastDB service.
- 2 Make a copy of the current repository content in:  
C:\Program Files\BIRTAnalytics\data\FastDB\databases
- 3 Delete the current repository files in:  
C:\Program Files\BIRTAnalytics\data\FastDB\databases
- 4 Copy the content of:  
C:\Financial  
to:  
C:\Program Files\BIRTAnalytics\data\FastDB\databases
- 5 Start the BIRT Analytics - FastDB service.
- 6 Synchronize and assign permissions using BIRT Analytics Administration, as described in “Synchronizing data” in Chapter 2, “Working with projects.”
- 7 Log in to BIRT Analytics and BIRT Analytics Loader to verify the restore. Refresh the repository view, if needed.

## Moving data objects to a different repository

You can move data objects from one repository to another by using the EXPORT and LOAD features of BIRT Analytics loader.

Use EXPORT to save a data object as a text file. Use LOAD to import the text file into a repository. For more information see *EXPORT* and Loading data.

It is also possible to load data objects using the Load tab in BIRT Analytics user interface without having to go to Loader to do it. You can also load directly from another instance of BIRT Analytics.

## A

### **access control list (ACL)**

A group or set of users with access to a database object. Using the BIRT Analytics Administration tool, the administrator creates a security group or ACL that manages privileges for a database object.

**Related terms**

BIRT Analytics Administration, column, database, group, security role, table

### **action**

An action is an event executed by a manual or task trigger. Example actions include send e-mail, query action, delete column, and apply model.

**Related terms**

scheduled task, trigger

### **Aggregates**

A tool that supports grouping data from multiple tables in one table. Aggregates supports defining a function and filter as properties.

**Related terms**

filter, table

### **analysis**

A tool that provides a specific view of data stored in FastDB. BIRT Analytics supports multiple analyses.

**Related terms**

Bubble analysis, Crosstab analysis, Evolution analysis, FastDB, Map analysis, Pareto analysis, Profile analysis, Venn analysis

### **antecedent**

Terms representing the left-hand, or If... clause of an association rule. The antecedent clause of an association rule contains discrete data items.

**Related terms**

association rules, consequent

### **association rules**

A predictive analytics technique that analyzes data for frequent If...Then patterns and calculates support and confidence criteria that identify the most important relationships. Support indicates how frequently the items appear in the database. Confidence indicates the number of times the If...Then relationships evaluate true.

An association rule has two parts, an antecedent and a consequent. The antecedent represents one or multiple data items. The consequent represents an item found in combination with the antecedent. An association rule returns a lift and a leverage value that measure how well the rule predicts the consequent.

**Related terms**

antecedent, Association Rules, confidence, consequent, lift, leverage, predictive analytics, support

## Association Rules

A predictive analytics tool that uses association rules to identify an If...Then relationship between data values stored in an information repository. For example, an association rule may show the following relationship: If a customer buys products A and B, then the customer also buys product C.

**Related terms**

association rules, predictive analytics

## B

### baseline filter

A filter that returns a group of records to serve as a basis for comparison. For example, use the year 2012 as a baseline filter for profit, to compare profit earned in another year with profit earned in 2012.

**Related terms**

filter, record

### big data analysis

The practice of analyzing, exploring, filtering, loading, segmenting, and studying massive quantities of data. Big data analysis uses statistics to describe qualities and predict trends in these data repositories.

**Related terms**

analysis, BIRT Analytics, data repository

### BIRT Analytics

An application, including a data repository, data loader, and web service, that supports big data analysis.

**Related terms**

big data analysis, BIRT Analytics Administration

### BIRT Analytics Administration

A BIRT Analytics system administration tool that runs as a browser-based application. The administrative user has full permission to modify all configurable features of the BIRT Analytics system.

**Related term**

BIRT Analytics

### BIRT Analytics Loader module

A tool that extracts, transforms, and loads records from an external data source to FastDB.

**Related terms**

BIRT Analytics, BIRT Analytics Administration, FastDB

### Bubble analysis

A tool that supports viewing a spatial distribution of data with respect to two axes.

**Related terms**

analysis, Crosstab analysis, Evolution analysis, Map analysis, Pareto analysis, Profile analysis, Venn analysis

## C

### calculated field

A data field that displays the result of an expression.

<b>campaign</b>	<p>A set of tasks, defined for specific population segment. A campaign is completed during a defined time period to accomplish a specific goal.</p> <p><b>Related term</b> segment</p>
<b>Canvas</b>	<p>A workspace for data analysis gadgets. Canvas supports arranging, assembling, and saving a collection of data visualization gadgets.</p> <p><b>Related term</b> gadget</p>
<b>cell</b>	<p>A set of properties that defines campaign actions to be performed for all records in a segment.</p> <p><b>Related terms</b> action, campaign, record, segment</p>
<b>Clustering</b>	<p>A predictive analytics tool that uses k-means cluster analysis. Clustering identifies groups of similar data values in large segments stored in a big data repository.</p> <p><b>Related terms</b> k-means, cluster analysis, predictive analytics</p>
<b>cluster analysis</b>	<p>A data analysis task that iterates estimating of values assigned to common data attributes. Common attributes identify groups of similar items, called clusters. Comparing clusters highlights similar and different groups in big data.</p> <p><b>Related terms</b> analysis, big data analysis, Clustering</p>
<b>column</b>	<p><b>1</b> A named field in a database table or query. For each data row, the column can have a different value, called the column value. The term column refers to the definition of the column, not to any particular value.</p> <p><b>2</b> A vertical sequence of cells in a crosstab, grid element, or table element.</p> <p><b>Related terms</b> column-oriented DBMS, database, data field, query, table</p>
<b>column-oriented DBMS</b>	<p>A column-oriented DBMS is a database management system (DBMS) that stores data tables as sections of columns of data rather than as rows of data. A column-oriented DBMS serializes all of the values of a column together, then the values of the next column, and so on.</p> <p><b>Related terms</b> database, column</p>
<b>confidence</b>	<p>An expression used to identify an association rule. Confidence compares how often the consequent appears when the antecedent is met. The confidence expression has the following syntax:</p> $\text{Confidence (A,B} \rightarrow \text{C)} = \text{Support (A,B,C)} / \text{Support (A,B)}$ <p><b>Related terms</b> association rules, support</p>
<b>consequent</b>	<p>Terms representing the left-hand, or ...Then clause of an association rule. The consequent clause of an association rule contains items found in combination with items in the antecedent.</p> <p><b>Related terms</b> antecedent, association rules</p>
<b>Convert</b>	<p>A BIRT Analytics option that displays results from one data analysis using a different type of data analysis. For example, an analysis created using Crosstab converts to a Bubble, Evolution, or Map analysis.</p>

**Related term**  
analysis

**count** The total number of records in a field.

**Related terms**  
field, record

### Crosstab analysis

A tool that supports analyzing data using cross-tabulation, or pivoting of different fields.

**Related terms**  
analysis, Bubble analysis, Evolution analysis, Map analysis, Pareto analysis, Profile analysis, Venn analysis

**Cylinder** A data visualization gadget that displays numeric values and boundaries in ranges. A Cylinder displays defined data measures as colored slices that comprise one cylinder shape.

**Related terms**  
Dial, Funnel, gadget, Gallery, Label, Meter, Sphere

## D

**data analysis** A process including acquiring, organizing, transforming, and modeling data to support decision-making.

### Data Explorer

A tool that displays records from a database stored in FastDB. Data Explorer provides a summary view for a table and a detail view for records, tables, selections, and segments.

**Related terms**  
Data Tree, FastDB, record, table

**data field** A location storing data having a specific type. A data field typically contains data from a database or other data source. A data field appears as a column when viewing a table in Data Explorer. For example, the BIRT Data Analytics Demo database includes the data field types listed in Table G-1.

**Table G-1** Data field types

Icon	Field type	Description
	Calculated	Displays a value result from an expression
	Date	Contains numbers that represent day, month, and year
	Date and time	Contains numbers that represent day, month, year, and time of day
	Full numeric	Contains whole, or integer numbers, such as 1 or 1000
	Real numeric	Contains real, or partial numbers such as 1.05 or 0.003
	Time	Contains a value representing time of day
	Text	Contains a string of alphabetic characters

**Related terms**  
record, Data Explorer, Data Tree, data types

**data integration**

A process through which data in varied sources is combined.

**data mining**

A computational process used to extract and transform data to prepare it for analysis.

**Related term**  
analysis

**data repository**

A physical or virtual location for storage and retrieval of data.

**Related term**  
FastDB

**Data Tree**

A tool that supports viewing and working with databases, tables, and records stored in FastDB. Data Tree includes Discrete Values, My Data, and My Folders viewers.

**Related terms**  
database, Discrete Values Viewer, My Data Viewer, My Folders Viewer, record, table

**data types**

A data type defines the limits of a data field in a BIRT Analytics database. For example, the BIRT Data Analytics demo database includes the data types listed in Table G-2.

**Table G-2** Data types in BIRT Analytics Loader

Data type	Description
Date	Contains numbers that represent day, month, and year. The default format is mm_dd_yyyy.
Datetime	Date and time data from January 1, 1753, through December 31, 9999, providing accuracy to three-hundredths of a second, or 3.33 milliseconds. The default format is yyyy_mm_dd_hh_MM_ss.
Integer	Integer data from $-2^{31}+1(-2,147,483,647)$ through $2^{31}-1(2,147,483,647)$ .
Longint	Integer data from $-2^{63}+1(-9,223,372,036,854,775,807)$ through $2^{63}-1(9,223,372,036,854,775,807)$ .
Real	Floating precision number data with the following valid values: $-1.79769 \times 10^{308}$ through $1.79769 \times 10^{308}$ .
String	A sequence of ASCII characters.
Time	Contains a value representing time of day. The default format is hh_MM_ss.
Unicode	A sequence of characters based on consistent encoding, representation, and handling of text as expressed in global writing systems.

**Related terms**  
Data Explorer, data field, Data Tree, record

**database**

**1** An integrated collection of logically related records that provides data for information application platforms, such as BIRT. The database model most commonly used is the relational model. Other typical models are entity-relationship, hierarchical, network, object, and object-relational.

**2** An integrated set of logically related records stored in FastDB.

**Related terms**  
record, table

**decision tree**

A predictive analytics technique that predicts the value of a target variable, based on values of multiple input variables. For example, use a decision tree to predict a survival rate, based on characteristics of the population that may survive.

**Related terms**  
Decision Tree, predictive analytics

### **Decision Tree**

A predictive analytics tool that uses the decision tree technique to predict an outcome, based on values of multiple input variables. For example, use Decision Tree to predict the product a customer will purchase, based on customer, purchase, gender, occupation, and income data.

**Related terms**  
association rules, predictive analytics

**Decodes** A tool that supports renaming a data field stored in FastDB.

**Related terms**  
data analysis, data field, FastDB

**Dial** A data visualization gadget that uses a needle-shaped pointer to display defined measures and numeric values in a range.

**Related terms**  
Canvas, Cylinder, Funnel, gadget, Gallery, Label, Meter, Sphere

### **Discrete Values Viewer**

A tool that supports viewing discrete values in a data record, selection, or segment.

**Related terms**  
My Data Viewer, My Folders Viewer, record, segment, selection

**Downloads** A tool that supports writing FastDB records to an external database.

**Related terms**  
database, Export file, FastDB, record

**Dubnium.exe** The file that runs the BIRT Analytics data repository, FastDB.

**Related term**  
FastDB

## **E**

### **Evolution analysis**

A tool that supports viewing a time-progression view of data values.

**Related terms**  
analysis, Bubble analysis, Crosstab analysis, Map analysis, Pareto analysis, Profile analysis, Venn analysis

### **Export Analytic DB**

A tool that supports creating a new database field based on a segment defined in the database. The new field is stored in FastDB.

**Related terms**  
Export file, FastDB, segment

**Export file** A tool that supports creating a new text file based on a segment defined in the database. The file is stored in FastDB.

**Related terms**  
Downloads, FastDB, segment

**Expressions** A tool that supports creating a logical relationship, using data fields, functions, and operators. Results of the relationship appear as a calculated field in FastDB.

**Related terms**  
calculated field, data field, FastDB

## F

- FastDB** The BIRT Analytics data repository. FastDB is a web service that caches data and supports executing data analysis and forecasting algorithms.  
**Related terms**  
Data Tree, database, data repository, record, table
- field** See data field.
- filter** A function that limits the number of records included a segment or selection. BIRT Analytics supports the following three filter types: baseline, target, and universal.  
**Related terms**  
baseline filter, target filter, universal filter
- Forecasting** A predictive analytics tool that uses the Holt-Winters, iterative method. Forecasting predicts a future trend in data exhibiting a seasonal pattern.  
**Related terms**  
Holt-Winters, predictive analytics
- functionalities**  
The system privileges an administrator grants to a security role.  
**Related terms**  
BIRT Analytics Administration, security role
- Funnel** A data visualization gadget displaying numeric values and boundaries that represent groups in a range, using colored bands that display on a funnel shape.  
**Related terms**  
Cylinder, Dial, gadget, Gallery, Label, Meter, Sphere

## G

- gadget** A computer program that provides services without requiring an application for each one. BIRT Analytics provides multiple gadgets that support data visualization.  
**Related terms**  
Cylinder, Dial, Funnel, Gallery, Label, Meter, Sphere
- Gallery** A tool that supports running multiple data visualization gadgets. Use the Gallery to assemble, arrange, and save gadgets on the Canvas.  
**Related terms**  
Canvas, Cylinder, Dial, Funnel, gadget, Label, Meter, Sphere
- group** A set of users belonging to the same organizational unit who share the same permissions for performing tasks. Using the BIRT Analytics Administration tool, the administrator creates a group from the list of available users on the system.  
**Related term**  
BIRT Analytics Administration

## H

- has seasonality**  
User-selected option that recognizes a seasonal trend in a data set.  
**Related terms**  
Holt-Winters, seasonal periodicity, seasonality
- Holt-Winters** A popular numerical estimation method used to forecast values in data that exhibit seasonal trends. The Holt-Winters method repeats and refines a time-series formula that includes a

level, trend, and seasonal component. The formula calculates forecast values valid for time  $t$  using a weighted average for all data prior to time  $t$ .

**Related term**

Forecasting

## I

**Import** A tool that supports adding a field to a database by uploading records from an external database. The field is stored in FastDB.

**Related terms**

database, FastDB, field

**indexed field** A data field having an associated key. An indexed field appears in a summary table used for data retrieval.

**Related terms**

data field, field, table

## J-K

**k-means** An iterative method of cluster analysis that groups large data sets into clusters of similar data. A k-means method forms clusters around data values having the nearest mean.

**Related terms**

analysis, Clustering, cluster analysis, mean

**kurtosis** A coefficient that describes the degree of concentration for a distribution of values, based on a mathematical average. The kurtosis coefficient is a value between -0.5 and 0.5. Colloquially, the Kurtosis coefficient is an average that indicates how sharp a distribution is with respect to a standard normal distribution.

**Related terms**

skewness, standard normal distribution

## L

**Label** A data visualization gadget that associates specific alphanumeric characters with a defined measure. A Label displays a text description of a measure in the BIRT Analytics Gallery.

**Related terms**

Canvas, Cylinder, Dial, Funnel, gadget, Meter, Sphere

**leverage** A value that indicates how well an association rule predicts the consequent. The method used to calculate leverage differs from the method used to calculate lift.

**Related terms**

association rules, lift

**lift** A value that indicates how well an association rule predicts the consequent. A lift value greater than one indicates that the items in the rule appear together more than expected. The method used to calculate lift differs from the method used to calculate leverage.

**Related terms**

association rules, leverage

**Links** A tool that supports maintaining links binding columns and tables in a database stored in FastDB.

**Related terms**

column, table

# M

## **make permanent**

A field operation that creates a new data field from either a calculated field or a current segment. The data field appears in FastDB.

### **Related terms**

calculated field, FastDB, field, segment

## **Map analysis**

A tool that supports plotting data values and regions on a geographic map. For example, a map analysis shows geographic regions and the number of high-net-worth customers in each region.

### **Related terms**

analysis, Bubble analysis, Crosstab analysis, Evolution analysis, Pareto analysis, Profile analysis, Venn analysis

## **maximum**

The highest registered value in a set of values.

### **Related term**

minimum

## **mean**

An arithmetic mean of all registered values in the field.

### **Related terms**

median, mode

## **median**

A value that divides a field into two symmetrical parts.

### **Related terms**

mean, mode

## **Meter**

A data visualization gadget that uses colored bars to display numeric values and boundaries in a range.

### **Related terms**

Canvas, Cylinder, Dial, Funnel, Gallery, Label, Sphere

## **minimum**

The lowest registered value in a set of values.

### **Related term**

maximum

## **mode**

The values having the most frequent number of occurrences in a field.

### **Related terms**

mean, median

## **My Data Viewer**

A tool that supports viewing fields and tables in multiple databases stored in FastDB.

### **Related terms**

Data Explorer, database, Discrete Values Viewer, field, My Folders Viewer, table

## **My Folders Viewer**

A tool that supports viewing reports, selections, and gadgets by a user or, if shared, by other users. My Folders appears as a tab in Data Tree and in the Start pane.

### **Related terms**

Data Explorer, Discrete Values Viewer, gadget, My Data Viewer, selection

## N

### NetScaler Web Logging (NSWL) query

A type of SQL query that tracks HTTP data traffic and writes information to a log file in a standard format such as the following example:

```
Select * from [Demo].[Household]
where [Demo].[Household].[Town]='LONDON';
```

**Related terms**

BIRT Analytics Administration, query, security filter, SQL (Structured Query Language)

### normal distribution

A bell-shaped, single-peaked, symmetric distribution of data. In a normal distribution, the mean, mode, and median coincide at the center.

**Related term**

standard normal distribution

### Numeric Ranges

A tool that supports creating a calculated field that includes a series of ranges into which data from numeric fields is grouped. For example, Numeric Ranges supports defining the following age ranges: Young - for age values less than 21, Adult - for age values 21 through 67, and Old - for age values greater than or equal to 67.

**Related terms**

calculated field, field

## O-P

### parameter

A variable expression that accepts a defined set of values.

**Related term**

filter

### Parametric

A tool that supports creating a field based on a defined condition, for use as a filter on a measure.

**Related term**

field

### Pareto analysis

A tool that supports comparing data using the Pareto principle, a commonly accepted rule which implies a data distribution with a numeric ratio of 80% to 20%. For example, the Pareto principle implies that 80% of sales result from 20% of customers.

**Related terms**

analysis, Bubble analysis, Crosstab analysis, Evolution analysis, Map analysis, Profile analysis, Venn analysis

### predictive analytics

A subject encompassing a variety of techniques used to analyze current and historical facts to make predictions about future, or otherwise unknown events. Credit scoring is a well-known application that uses predictive analytics techniques to generate a score for an individual, based on credit history data for that individual.

**Related terms**

Association Rules, Clustering, Decision Tree, Forecasting

### Profile analysis

A tool that supports identifying a set of similar characteristics in a group. A profile analysis compares z-score values calculated for each set of characteristics.

**Related terms**

analysis, Bubble analysis, Crosstab analysis, Evolution analysis, Map analysis, Pareto analysis, Venn analysis, z-score

**profile**

A set of associated security roles, groups, filters, and users. Using the BIRT Analytics Administration tool, the administrator creates a profile from the lists of roles, groups, filters, and users available on the system. From the BIRT Analytics security options list, choose Profiles, specify a profile name, provide a description, then select the roles, groups, filters, and users to include in the profile.

**Related terms**

BIRT Analytics Administration, group, security filter, security role

**prompted filter**

A data set filter that supports user entry of parameter values.

**Related terms**

filter, parameter

## Q

**Quantile**

A tool that supports creating a new calculated field by grouping values in a numeric field, using multiple groups that contain an equal number of values. For example, use Quantile to group a field containing 2400 values into four quartiles having 600 values each.

**Related terms**

calculated field, field

**query**

A statement specifying the data rows to retrieve from a data source. For example, a query that retrieves data from a database typically is a SQL SELECT statement.

**Related terms**

database, SQL (Structured Query Language)

## R

**Ranking**

A tool that supports ordering a table by generating a column of calculated values that correspond to a sorted column. The calculated values represent an ordered list of ranks.

**Related terms**

column, table

**record**

A set of related, indexed data fields in a database. A record often appears as a row shown in a table. For example, a customer record could include a numeric field for customerID, a character string field for customer name, and an alphanumeric field for age group.

**Related terms**

field, row

**row**

See record.

**Related terms**

field, record

## S

**scheduled task**

A scheduled task includes a trigger, task details, and an assigned action.

**Related terms**

action, campaign, stage, trigger, workflow

<b>Scratchpad</b>	<p>A BIRT Analytics work area that supports temporary caching of multiple segments. Scratchpad also supports creating new fields based on segments or selections.</p> <p><b>Related terms</b> Data Explorer, Data Tree, segment, selection</p>
<b>seasonal periodicity</b>	<p>A value indicating the number of periods in a cycle. Input a value for seasonal periodicity to initiate a forecast that predicts a seasonal pattern in a data set.</p> <p><b>Related terms</b> Forecasting, Holt-Winters</p>
<b>seasonality</b>	<p>In a data set, a periodic trend that corresponds to monthly, quarterly, or semi-annual periods such as seasons.</p> <p><b>Related terms</b> Forecasting, Holt-Winters</p>
<b>security role</b>	<p>A set of functionalities that an administrator uses to configure permissions in the BIRT Analytics system.</p> <p><b>Related terms</b> BIRT Analytics Administration, functionalities, query, security filter</p>
<b>security filter</b>	<p>A type of query that an administrator uses to limit access to data in the BIRT Analytics system.</p> <p><b>Related terms</b> BIRT Analytics, BIRT Analytics Administration, group, NetScaler Web Logging (NSWL) query</p>
<b>segment</b>	<p>A segment is a group of records sharing at least one common characteristic.</p> <p><b>Related terms</b> record, selection</p>
<b>selection</b>	<p>A selection is a user-specified request that returns a segment from a database.</p> <p><b>Related terms</b> record, segment</p>
<b>skewness</b>	<p>A value that reflects the distribution of values in a data set. Skewness values can be positive, zero, or negative. A positive value reflects a data set in which more values lie to the left of the mean value. A negative value reflects a data set in which more values lie to the right of the mean. A zero value indicates values distributed evenly around the mean, typically implying a symmetric distribution.</p> <p><b>Related terms</b> kurtosis, mean</p>
<b>Sphere</b>	<p>A data visualization gadget that uses a colored sphere shape to display numeric values and boundaries in a range.</p> <p><b>Related terms</b> Canvas, Cylinder, Dial, Funnel, gadget, Label, Meter</p>
<b>SQL (Structured Query Language)</b>	<p>A language used to access and process data in a relational database.</p> <p><b>Related term</b> database</p>
<b>stage</b>	<p>A tool that supports defining users as task owners and assigning to each task owner the permissions required to perform tasks. Define a stage to identify part of a campaign.</p> <p><b>Related terms</b> action, campaign, scheduled task, trigger, workflow</p>

**standard deviation**

The value equal to the positive square root of variance calculated for a data set.

**Related term**  
variance

**standard normal distribution**

The normal distribution in which the mean is zero and the standard deviation is one.

**Related term**  
normal distribution

**Standardize column**

A tool for preprocessing data values having a distribution different from a standard normal distribution. Multiple options support value sets distributed closely, clustered, spread, or having many repeated values.

**Related term**  
standard normal distribution

**sum**

The cumulated sum of all the values in a field.

**Related term**  
sum-of-squares

**sum-of-squares**

The sum of all of the squared values in a set.

**Related term**  
sum

**support**

An expression that calculates a ratio measuring how many transactions contain all items in an association rule. The support expression has the following syntax:

`Support (A,B) = Transactions (A,B)/Total transactions`

**Related term**  
association rules

**T****table**

A named set of records in a database.

**Related terms**  
database, record

**target filter**

A filter that returns a group of records for comparison with an established baseline. For example, use the year 2010 as a target filter for profit, to compare profit earned in 2010 with profit earned in another, baseline year.

**Related terms**  
baseline filter, universal filter

**temporal file**

A temporary data file generated and stored in the system cache. Using the BIRT Analytics Administration tool, the administrator can remove the accumulated temporal files and records created by an application to optimize performance.

**Related term**  
BIRT Analytics Administration

**trigger**

A trigger is a time or event that starts a scheduled task.

**Related terms**  
action, campaign, scheduled task, stage, workflow

## U

### universal filter

A filter that is always applied at a lower resolution level, before changing resolution.

**Related terms**

baseline filter, target filter

## V-Y

### value

1 The content of a constant, parameter, symbol, or variable.

2 A specific occurrence of an attribute. For example, blue is a possible value for an attribute color.

**Related term**

parameter

### variance

A value equal to the squared average of the distances between each value and the arithmetic mean.

**Related term**

mean

### Venn analysis

A tool that supports data analysis based on crossing more than two fields. A Venn analysis identifies coincident values in multiple data segments. For example, use a Venn analysis to show how many customers buy the same three products.

**Related terms**

Bubble analysis, Crosstab analysis, Evolution analysis, Map analysis, Pareto analysis, Profile analysis, segment

## W

### workflow

A role responsible for completing tasks or stages in a campaign.

**Related terms**

action, campaign, scheduled task, stage, trigger

## Z

### z-score

A value describing whether a quantifiable difference between two groups is statistically significant.

**Related term**

Profile analysis

# Index

## Symbol

- , (comma) character 105
- \* operator 66, 104
- / (slash) character 104
- / operator 66
- + operator 66
- (hyphen) character 104
- operator 66

## A

- ABS function 67
- access control list 113
- access control lists 15
- access permissions. *See* privileges
- access restrictions 13, 18, 105
- accessing
  - BIRT Analytics Admin application 102
  - BIRT Analytics Loader 3
  - data 22
  - documentation vii
  - projects 10
  - text files 29
  - Transform text editor 59
- accounts 4, 12, 18, 103
- ACLs. *See* access control lists
- actions 35, 113
- Active property 105
- Actuate BIRT Analytics
  - See also* BIRT Analytics
- Actuate Customer Support viii
- Actuate product information viii
- adding
  - aggregate columns 60
  - calculated columns 66
  - calculated fields 123
  - data fields 122
  - data sources 23
  - decoding columns 61, 62
  - domains 63, 64
  - links 53, 85
  - load processes 32–33
  - parametric columns 78
  - quantile range columns 80
  - ranking columns 81, 82
  - remote data providers 29
  - user accounts 18
  - users 12, 102
- Admin application. *See* BIRT Analytics Administration
- Admin module (Loader) 18
- Admin tab (Loader) 12
- Admin tools 114
- Administration page (Loader) 102

- Administration tool. *See* BIRT Analytics Administration
- administrator accounts 4
- Administrator property 103
- administrators 8, 102
- AGE function 67
- Aggregate dialog box 60
- aggregate filters 60
- AGGREGATE instruction 60
- Aggregates tool 113
- aggregation functions 60, 61
- ALLMONTHS function 67
- ALLQUARTERS function 67
- analysis (defined) 113
- analysis gadgets. *See* gadgets
- analytical functions 59
- Analytics Admin tool. *See* BIRT Analytics Administration
- Analytics application. *See* BIRT Analytics
- Analytics Loader. *See* BIRT Analytics Loader
- analytics processes
  - See also* ETL processes
- analyzing data 13
- AND operator 66
- antecedent 113
- Association Rules 114
- association rules 113
- Attributes tab (Explorer) 50
- authentication 18
- automating
  - load processes 107
  - synchronization processes 16
- Average function 60

## B

- backing up FastDB database 110
- Bands parameter 76
- baseline filters 114
- batch processes 105
- big data 2
- bins 78, 83
- Bins parameter 78, 83
- BIRT Analytics vi, 3
- BIRT Analytics Administration 17, 102
- BIRT Analytics Administration tool 114
- BIRT Analytics application 114
- BIRT Analytics Loader 2, 3, 5, 114
- BIRT Analytics Loader command-line tool 16
- BIRT Analytics Loader configurations 108
- BIRT Analytics Loader engine 107
- BIRT Analytics Loader features 12
- BIRT Analytics Loader home page 4
- BIRT Analytics Loader login page 3
- BIRT Analytics Loader port 10

BIRT Analytics Loader user accounts 3  
BIRT Analytics Loader user interface 11, 103  
Boolean values 66  
bubble analysis (defined) 114

## C

calculated columns. *See* calculated fields  
calculated field type 116  
calculated fields 59, 66, 75, 114, 123  
calculated values 123  
Canvas tool 115  
CEIL function 67  
changing  
    column definitions 41  
    data repository 16  
    data sources 23  
    database names 88  
    database records 16  
    FastDB database 112  
    load processes 33  
    load transactions 34  
    passwords 12, 18  
    project names 12  
    projects 11  
    remote data providers 29  
    scheduled tasks 105  
    transformation scripts 56  
    user information 103  
character sets 36  
CHR function 67  
closing  
    BIRT Analytics Loader sessions 5  
    database tables 41  
    projects 11  
cluster analysis 115  
Clustering tool 115  
CODE function 67  
Code position parameter 62  
column bindings. *See* links  
Column declaration dialog box 38  
column definitions 38, 41, 58  
Column name parameter 82, 83, 87  
column names 61, 87, 88  
column offsets 37  
Column to use parameter 82  
column transformation instructions 59  
column-oriented DBMS 115  
columns  
    *See also* calculated fields; fields  
    adding to target tables 38  
    aggregating values for 60  
    changing properties for 41  
    changing values in 60  
    creating calculated 66, 75  
    creating dynamic 52, 59  
    creating parametric 78  
    decoding 61  
    defined 115

    displaying 49, 59  
    excluding from load processes 39  
    exporting 89  
    extracting from source files 37, 42  
    grouping values for 76, 80  
    linking to 12, 85  
    making permanent 52, 86  
    ranking record subsets and 81  
    removing from tables 39, 83  
    renaming 61, 88  
    sorting values in 64, 82  
    validating 37, 87  
    viewing properties for 50, 51  
command-line options (qLauncher) 106  
command-line tools (BIRT Analytics) 16  
comma-separated values files 29, 39  
comments (project log files) 97  
comments (transformation scripts) 58  
computed columns. *See* calculated fields  
computed fields. *See* calculated fields  
concatenation 66  
concentration (values) 120  
conditional expressions 66  
confidence 115  
configuration parameters 12  
    *See also* parameters  
configurations 22, 108  
connection information 22, 25  
connection properties 25, 26, 27  
connections  
    accessing data and 3, 22, 29  
    creating projects and 10  
    running native drivers and 26  
    running non-native drivers and 27  
    setting up 12  
    validating 40  
consequent 115  
constants 66  
control menus 11  
Convert tool 115  
copying  
    FastDB database 111  
    load projects 109  
    repository databases 112  
COUNT function 67  
Count function 60  
count operations 116  
Create new project dialog 9  
creating  
    aggregate columns 60  
    calculated columns 66  
    calculated fields 123  
    data fields 122  
    data sources 23  
    decoding columns 61, 62  
    domains 63, 64  
    dynamic columns 52, 59  
    expressions 75

- FastDB database 110
- links 53, 85
- load processes 32–33
- load transactions 34
- mapping specifications 34, 42
- numeric ranges 76
- parametric columns 78
- projects 9–11
- quantile range columns 80
- queries 42, 43
- ranking columns 81, 82
- remote data providers 29
- scheduled tasks 104
- target data tables 38
- transformation script files 56
- transformation scripts 57
- transformation transactions 56
- user accounts 18
- users 12, 102
- credential files 105, 106
- credential tokens 106
- credentials 4, 18, 105
- cron 101
- Crosstab analysis (defined) 116
- CSV files 29, 39
- custom data sources. *See* ODBC data sources
- Customer Support viii
- cylinder gadgets 116

## D

- data
  - See also* values
  - accessing 22
  - analyzing 13
  - controlling access to 13, 18
  - grouping 60, 63, 76, 80
  - limiting access to 124
  - manipulating 56
  - protecting sensitive 3, 12, 105
  - retrieving 42, 44
  - synchronizing 13, 16, 17
  - updating 9
  - uploading 29, 31
  - viewing 12
- data analysis 116
- data analysis gadgets. *See* gadgets
- Data Explorer 116
- data export operations 89, 91, 92
- data extraction 2, 8, 22, 37
- data field names 61, 88
- data field types 116
- data fields
  - See also* columns
- data fields. *See* fields
- data filters. *See* filters
- data load operations. *See* load operations
- data load processes. *See* load processes
- Data Management module 48

- Data Management tab 12
- data mapping models 34
- data mining 2
- data modeler 8
- data object transformation instructions 83
- data objects 83
- data providers 29, 31
- data repository vi, 9, 118
  - See also* FastDB database
- data rows 49, 50, 123
  - See also* records
- data set filters. *See* filters
- data sets 42, 44, 78, 83
- data source connections. *See* connections
- data source mapping specifications 12, 22, 34, 42
- data sources 3, 8, 23
- Data tab (Loader) 12, 22
- data transformation 2, 8, 56
- data transformation rules 12
- Data Tree 117
- data types 38, 50, 117
- database connection information 22, 25
- database connections 22, 26, 27
- database drivers 22, 25
- database loading processes. *See* load processes
- database mapping specifications 42
- database model 117
- Database name parameter 84
- database names 88
- database objects 49, 94
- database repository. *See* FastDB database
- database structures 48
- database tables. *See* tables
- database types 22
- databases 9, 22, 112
  - defined 117
  - See also* data sources
- Databases panel (Data) 22
- Datasource dialog box 26, 27
- date and time field type 116
- date data type 117
- date field type 116
- DATE function 67
- datetime data type 117
- DATETIME function 67
- DAY function 67
- DAYADD function 67
- DAYSTO function 67
- decimal values 39
- Decision Tree 118
- decision tree 117
- Decode dialog box 62
- Decode position parameter 62
- Decodes tool 118
- decoding columns 61
- decoding files 61
- decryption 106
- default data repository 9

- deleting
  - columns 39, 83
  - data sources 23, 29
  - links 84
  - load transactions 34
  - scheduled tasks 105
  - script files 57
  - tables 85
  - transformation instructions 58
  - transient links 53, 54
  - user accounts 18
  - users 103
- delimited text files 37
- delimiter characters 37, 62
- Delimiter parameter 62
- Demo database 110
- derived column definitions 59
- derived columns
  - See also* calculated fields
- Description parameter (aggregation) 60
- Description property (scheduled tasks) 105
- developers vi, 8
- dial gadgets 118
- directory paths 109
- discrete values 50
- Discrete Values Viewer 118
- Discretes tab (Explorer) 51
- displaying
  - columns 49, 59
  - data 12
  - database tables 48
  - databases 48
  - discrete values 51
  - load process log files 12, 15, 105
  - project log files 14
  - project names 11
  - script files 96
  - transformation instructions 57
- DISTINCT function 67
- documentation vi
  - administering iHub System and vii
  - downloading vii
  - updating vii
- Domain definition dialog box 64
- DOMAIN instruction 63
- domain names 64
- Domain parameter 64
- domains 63, 88
- downloading
  - documentation files vii
- Downloads tool 118
- driver names 27
- drivers 3, 22, 25, 27
- DROPCOLUMN instruction 83
- DROPDATABASE instruction 84
- DROPLINK instruction 84
- DROPTABLE instruction 85
- Dubnium database 108

- Dubnium servers 109, 110
- Dubnium service 3
- Dubnium service security accounts 3, 18
- dubnium.exe 118
- dynamic columns 52, 59, 87

## E

- edit panel (table declarations) 42
- editable columns 52
- editor (Transform) 59
- Encoding parameter 62
- encoding types 36
- encryption 106
- End line parameter 62
- end-of-line characters 36, 62
- EQ operator 66
- estimation method 119
- ETL (defined) 2
- ETL developers vi, 8
- ETL processes 2, 8
  - See also* specific process
- events 113
- evolution analysis (defined) 118
- EXP function 67
- Explorer 48–49
- Export Analytic DB tool 118
- Export dialog box 89, 91, 93
- Export File tool 118
- EXPORT instruction 88
- export operations 89, 91, 92
- Export target and name parameter 88
- export tools 118
- Expression dialog box 75
- Expression parameter 66
- expressions 66, 75
  - defined 118
- External Call page (Administration) 105
- extraction processes 2, 8, 22, 37

## F

- FastDB database
  - adding databases to 112
  - backing up 110
  - changing 16, 112
  - copying 111
  - creating 110
  - installing 109
  - loading data into 9
  - removing databases from 84
  - removing objects from 49
  - renaming data fields in 61, 88
  - renaming databases in 88
  - setting up security accounts for 18
  - testing for objects in 94
  - viewing database structures in 48
- FastDB engine 3
- FastDB engine user accounts 3, 12, 18
- FastDB web service 119

- features vii, 12
- field (defined) 116
- field names 61, 88
- field types 116
- fields
  - adding calculated 123
  - building summary tables and 120
  - creating 122
  - See also* columns
- file connections 29
- file encoding types 36
- file headers 37
- file lists 32
- file names 32
- File parameter 61
- file paths 109
- file systems 29
- file types 35
- File Upload dialog box 29
- files
  - accessing data in 29
  - accessing help vii
  - copying project 109
  - creating script 56
  - data extraction and multiple 31
  - defining mapping specifications for 34, 39
  - downloading documentation vii
  - exporting data to 88, 91, 92
  - generating credential 105
  - opening script 57
  - selecting formatting properties for 35
  - transforming data and 56
  - uploading 29, 30, 32, 61
  - viewing log 12, 14, 15, 105
- Files panel (Data) 22, 29
- Filter parameter
  - AGGREGATE instruction 60
  - EXPORT instruction 89
  - INUMERIC instruction 76
  - QUANTILE instruction 80
  - RANKING instruction 82, 83
- filter types 119
  - See also* specific type
- filtering
  - record groups 114
- filters 119
- final links 53
- First function 60
- flat file data sources 29
- flat files 22, 37
  - See also* data sources
- FLOOR function 67
- Force sequence parameter 82
- Forecasting tool 119
- formatting properties (source files) 35
- formula names 52
- formulas 52, 60
  - See also* expressions
- full numeric field type 116
  - See also* numeric fields
- Function parameter 60
- functionalities 119
- functions 59, 67
- funnel gadgets 119

## G

- gadget types 119
- gadgets 119
- Gallery 119
- GE operator 66
- geographic maps 121
- grouping
  - records 124
- grouping data 60, 63, 76, 80
- groups (users) 15, 119
- GT operator 66

## H

- has seasonality option 119
- help (qLauncher) 106
- help files vii
- Holt-Winters method 119
- home page (Loader) 4
- HOUR function 67
- HTML documentation vii

## I

- IF function 67
- If statements 66
- IFEXISTS instruction 94
- IFLINKED instruction 94
- IIS services 105
- import operations 120
- importing metadata 41
- inactive transformation instructions 77
- Include dialog box 96
- INCLUDE instruction 96
- INCLUDE statements 94, 96
- Includes panel (Transform) 56
- indexed fields 120
  - See also* summary tables
- indexes 38, 50
- information. *See* data
- INSTR function 67
- INT function 67
- integer data type 117
- Internet Information Services. *See* IIS services
- INUMERIC instruction 76
- inversion 64
- IP addresses 3, 10
- IP parameter 10
- ISNOTNULL function 68
- ISNULL function 67

## K

- key pairs (encryption) 106
- Key to rank parameter 82
- KURT function 68
- Kurtosis 120
- Kurtosis coefficient 120

## L

- label gadgets 120
- Language property 103
- Last function 60
- LCASE function 68
- LE operator 66
- LEFT function 68
- LEN function 68
- leverage 120
- lift 120
- Link dialog box 85
- LINK instruction 85
- links 84, 85, 94
- Links tool 48, 52–54, 120
- load operation types 35
- load processes
  - automating 107
  - building mapping models for 34
  - building projects for 3, 8, 9
  - configuring 22
  - creating 32–33
  - defining connections for 22, 25, 29
  - running 2, 11
  - saving changes to 33
  - scheduling 103
  - setting options for 33
  - starting 13
  - viewing log files for 12, 105
- load projects. *See* projects
- load transactions
  - creating 34
  - defining source file formats for 35, 39
  - defining table columns for 37
  - defining target objects for 34, 42
  - loading data and 22, 33, 44
  - running 33
- Loader module 114
- Loader module. *See* BIRT Analytics Loader
- loading processes. *See* load processes
- local file systems 29
- log files 12, 14, 15, 97, 105
- LOG function 68
- LOG10 function 68
- logging load operations 33
- login credentials 4
- login names 103
- login page (Analytics Loader) 3
- Login property 103
- Logout icon (Loader) 11
- Longint data type 117

- LONGINT function 68
- LT operator 66
- LTRIM function 68

## M

- Main file link 57
- make permanent operations 121
- MAKEPERMANENT instruction 86
- map analysis (defined) 121
- mapping 34
- mapping specifications 12, 22, 34, 42
- mathematical operators 66
- MAX function 68
- Maximum value function 60
- maximum values 121
- MEAN function 68
- Mean function 60
- mean values 121
- MEDIAN function 68
- median values 121
- menus 11
- metadata 41
- meter gadgets 121
- Microsoft Internet Information Services. *See* IIS services
- Microsoft SQL Server databases 22
- MID function 68
- MIN function 68
- Minimum value function 60
- minimum values 121
- MINUTE function 68
- MODE function 69
- mode values 121
- MONTH function 69
- My Data Viewer 121
- My Folders Viewer 121
- MySQL databases 22, 25

## N

- Name parameter 10
- naming
  - database connections 25, 26
  - decoding columns 62
  - domains 64
  - FastDB data fields 61, 88
  - file data sources 32
  - ODBC database connections 28
  - parametric columns 78
  - projects 10
  - quantile range columns 81
  - ranking columns 82
  - variables 97
- native drivers 22, 25
- NE operator 66
- negative values 124
- NetScaler Web Logging queries 122
- New file list dialog box 32
- normal distribution 122

- NOT operator 66
- NOW function 69
- NSWL queries 122
- null values 50
- Number of parts parameter 80
- numeric field types 116
- numeric ranges 76, 77
- Numeric ranges dialog box 76
- Numeric Ranges tool 122
- numeric values 76, 80
- numerical estimation method. *See* Holt-Winters method

## O

- obsolete product information viii
- ODBC connectors 3
- ODBC data sources 27
- ODBC driver names 27
- ODBC drivers 27
- online documentation vi
  - administering iHub System and vii
  - downloading vii
  - updating vii
- opening
  - BIRT Analytics Admin application 102
  - BIRT Analytics Loader 3
  - BIRT Analytics Loader projects 10
  - script files 57
  - Transform text editor 59
- operators 66, 104
- Options panel (Data) 33
- OR operator 66
- Oracle databases 22, 42
- output files (qLauncher) 107

## P

- parameters
  - adding parametric columns and 83, 86
  - adding quantile range columns and 80
  - adding ranking columns and 82
  - aggregating data values and 60
  - changing project configurations and 12
  - defined 122
  - defining numeric ranges and 76
  - defining variables and 97
  - dropping database objects and 83, 84, 85
  - entering values for 123
  - exporting domains and 88
  - linking tables and 85
  - opening Loader module and 3
  - running INCLUDE statements and 94
  - running qLauncher utility and 106
  - testing for links and 94
- parametric columns 78
- Parametric dialog box 78
- parametric filters 122
- PARAMETRIC instruction 83, 86
- Parametric tool 122

- Pareto analysis (defined) 122
- Pareto principle 122
- Password parameter 10
- Password property 103
- passwords 4, 12, 18, 103
- paths 109
- PDF documentation vii
- periodic trends 124
- permanent columns 52, 86
- permissions. *See* privileges; functionalities
- Port parameter 10
- positive values 124
- PostgreSQL databases 23
- POWER function 69
- predictive analytics 122
- private encryption keys 106, 107
- privileges 13, 15, 110, 113, 119
- processes. *See* ETL processes
- product information viii
- product updates vii
- profile (users) 123
- profile analysis (defined) 122
- project development page 11
- project files 109
- project folders 108, 109
- project log files 14, 97
- project names 10, 11
- Project property 104
- projects
  - building ETL processes and 3, 8
  - creating 9
  - limiting log entries for 14
  - managing 11
  - naming 10
  - running 13, 106, 107
  - saving 11
  - storing 108
- Projects management page 11, 102
- prompted filter 123
- PROPER function 69
- PROPERSENTENCE function 69
- protecting data 3, 12, 105
- public encryption keys 106, 107
- public-key encryption 106

## Q

- qLauncher utility 106, 107
- qualifier characters 37, 62
- Qualifier parameter 62
- Quantile dialog box 80
- QUANTILE instruction 80
- quantile range columns 80
- quantile range filters 80
- Quantile tool 123
- QUARTER function 69
- queries
  - creating 42, 43
  - defined 123

- defining bins and 79
- defining domains and 64, 65
- limiting data access and 124
- retrieving data and 42, 78
- retrieving data with 123
- tracking HTTP data traffic and 122

Query columns parameter 89

query language 124

Query resolution table parameter 89

## R

RAND function 69

range of values 76, 77

ranking 81

ranking columns 81

Ranking dialog box 82

RANKING instruction 82

ranking keys 82

ranking sequence 82

Ranking tool 123

real data type 117

REAL function 69

real numeric field type 116

- See also* numeric fields

records

- changing 16
- creating sample of 64
- defined 123
- grouping 63, 124
- linking tables and 52
- ranking subsets of 81
- setting number of 33
- updating 16
- validating content 37

REGEXMATCH function 69

REGEXREPLACE function 69

relational databases 22

release notes vii

remote data providers 29, 31

removing

- columns 39, 83
- data sources 23, 29
- links 84
- load transactions 34
- scheduled tasks 105
- script files 57
- tables 85
- transformation instructions 58
- transient links 53, 54
- user accounts 18
- users 103

RENAME instruction 88

renaming

- databases 88
- FastDB database fields 61, 88
- projects 12

REPLACE function 69

REPORT instruction 97

reports 15

repository vi, 9, 118

- See also* FastDB database

Repository list (Transform) 58

Repository tree 48, 49, 50

resolution 64

RIGHT function 69

roles 124, 126

ROW function 69

rows 49, 50, 123

- See also* records

RTRIM function 69

rules (data transformation) 12

running

- Analytics Loader projects 13, 14, 105, 106, 107
- load processes 2, 11
- load transactions 33
- transformation instructions 58, 77
- transformation scripts 56, 94, 97

Running project log 14

## S

saving

- load processes 33
- projects 11

Scheduled task dialog 104

scheduled tasks 123

Scheduled tasks page (Administration) 103

scheduling load processes 103

Scratchpad 124

script files 56, 57, 96

scripts. *See* transformation scripts

seasonal patterns 124

seasonal periodicity 124

seasonal trends 119

seasonality 124

SECOND function 69

SECONDDSTO function 70

SECSTO function 70

secure connections 10

security 3, 12, 105

security accounts (Dubnium service) 3, 18

security filters 124

security groups 15

security keys 106, 107

security roles 124

segment blocks 64

segments 63

- defined 124

selections 124

server names 3

servers 105

Settings property 104

Settings tab (Loader) 12

SGN function 70

sharing connections 22

SKEW function 70

skewness operations 124

- skewness values 124
- Sorting direction parameter 82
- Source column parameter 60, 61, 76, 80
- source files 35, 37
- sphere gadgets 124
- spreadsheets 29
- SQL (defined) 124
- SQL queries. *See* queries
- SQL Server databases 22
- SQRT function 70
- Square sum function 60
- squared values 125
- SSL parameter 10
- stage 124
- standard deviation 125
- Standard deviation function 60
- standard normal distribution 125
- Standardize column tool 125
- starting
  - load processes 13
  - synchronization processes 16
- STDEV function 70
- stopping
  - Dubnium servers 109
- string data type 117
- STRING function 70
- STRIP function 70
- sum (defined) 125
- SUM function 70
- Sum function 60
- summary values 125
- sum-of-squares (defined) 125
- SUMSQ function 70
- support 125
- Support Lifecycle Policy viii
- Supported Products Matrix viii
- symmetric distribution 124
- synchronizing data 13, 16, 17
- system administration tool. *See* BIRT Analytics Administration
- system administrators. *See* administrators
- system privileges 119
- SYSTEM user 4

## T

- Table declaration dialog box 35, 38, 42
- table definitions 8, 12, 34
- Table name parameter 84, 85
- tables
  - adding dynamic columns to 59
  - adding fields to 122
  - binding columns to. *See* links
  - closing 41
  - defined 125
  - deleting 85
  - exporting 89
  - grouping data from multiple 60
  - grouping records in. *See* segments

- linking to 48, 52–54, 84, 85
- mapping source file data and 34
- ranking 81
- removing columns in 39, 83
- renaming 88
- selecting 42
- setting sorting direction for 82
- testing for links in 94, 95
- updating 9, 17
- viewing columns in 49, 59
- viewing list of 48
- tabs 12
- Target column parameter
  - AGGREGATE instruction 60
  - DECODE instruction 61
  - DOMAIN instruction 64
  - EXPRESSION instruction 66
  - INUMERIC instruction 76
  - PARAMETRIC instruction 78, 83
  - QUANTILE instruction 80
- target filters 125
- target objects 34, 42
- Target table parameter
  - AGGREGATE instruction 60
  - DOMAIN instruction 64
  - EXPRESSION instruction 66
  - PARAMETRIC instruction 78, 83
- technical support viii
- temporal files 125
- temporary files 125
- text editor (Transform) 59
- text field type 116
- text file data sources 29
- text files 37, 91
  - See also* data sources
- Text parameter 97
- time data type 117
- time field type 116
- TIME function 70
- time settings (scheduled tasks) 104
- TODAY function 70
- tokens (credentials) 106
- transactions 56
  - See also* load transactions
- Transform tab (Loader) 12
- transformation instructions 57, 77, 88
  - See also* transformation scripts
- transformation processes 2, 8, 56
- transformation rules 12
- transformation script files 56, 57, 96
- transformation scripts 33, 56, 57, 94, 97
- transformation transactions 56
- transient files 125
- transient links 53
- triggers 125
- TRIM function 70

## U

- UCASE function 70
- Unicode data type 117
- universal filters 126
- updates (documentation) vii
- updates (product) vii
- updating
  - database tables 9, 17
  - Includes file list 57
- uploading
  - data 29, 31
  - decoding files 61
  - local files 29
  - multiple files 32
  - remote files 30
- URI dialog box 31
- URI property 31
- URIs 31
- URLs
  - Actuate product information viii
  - Actuate technical support viii
  - BIRT Analytics Loader 3
  - release notes vii
- user accounts 4, 12, 18, 103
- user credentials 4, 18, 105
- user groups 15, 119
- user interface 11, 103
- user names 4
- User parameter 10
- user privileges 13, 15, 110
- users 3, 12, 102
- user-specified requests. *See* selections

## V

- values
  - aggregating 60
  - defining range of 76, 77
  - getting from multiple tables 78
  - grouping 76, 80
  - inverting 64
  - loading 52
  - ranking 123
  - viewing discrete 51
  - viewing number of discrete 50
  - viewing number of null 50
- VAR instruction 97
- Variable dialog box 98
- variables 97
- variance 126
- Venn analysis (defined) 126
- viewers 121
- viewing
  - columns 49, 59
  - data 12
  - database tables 48
  - databases 48
  - discrete values 51

- load process log files 12, 15, 105
- project log files 14
- project names 11
- script files 96
- transformation instructions 57

## W

- web addresses 30
- web browsers 3, 12
- web servers 105
- web services vi
- web services. *See* BIRT Analytics application; FastDB
  - web service
- WEEKDAY function 71
- workflow 126
- workflow roles 126

## Y

- YEAR function 71

## Z

- zero values 124
- Z-score operations 126