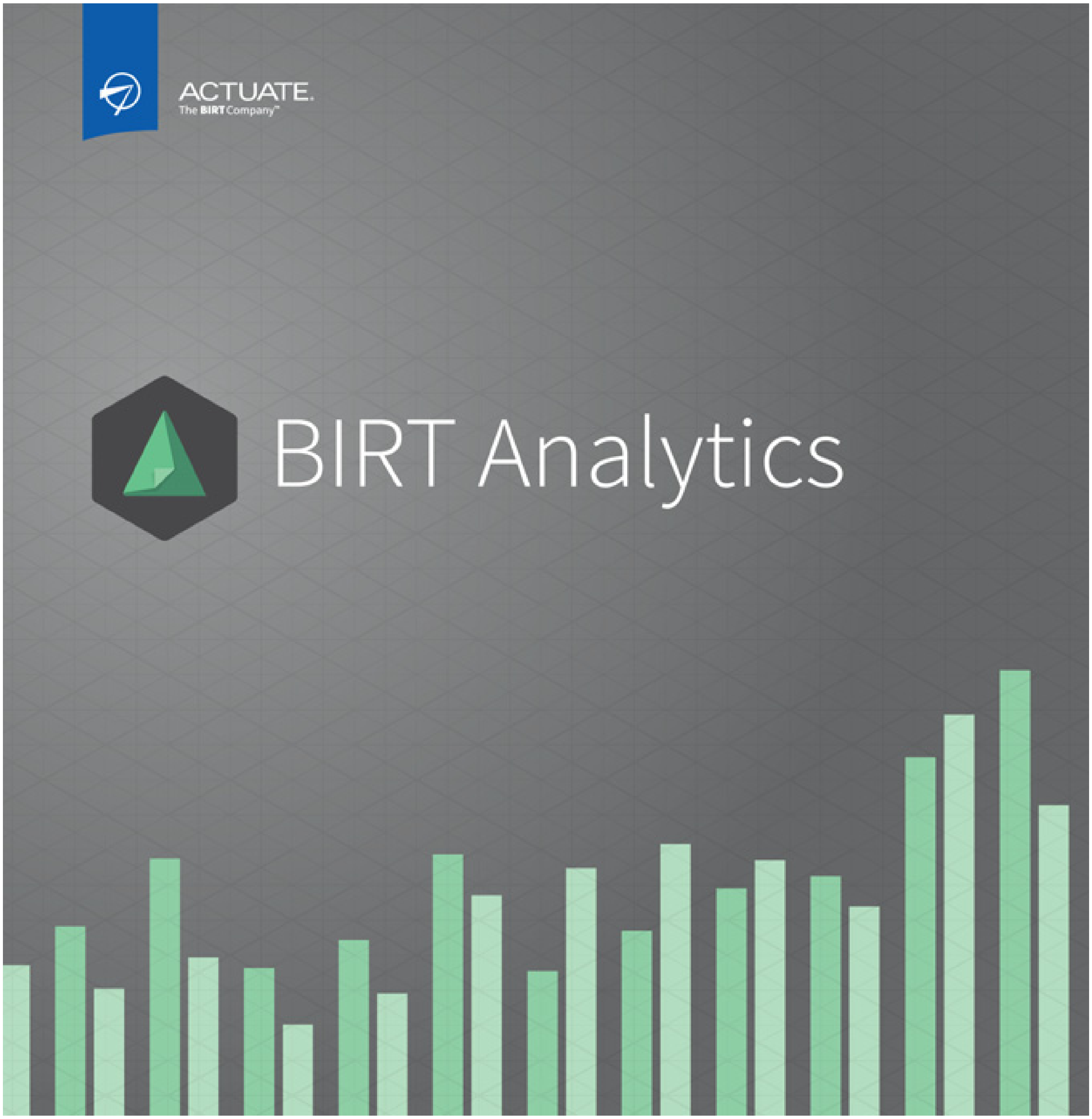


Using BIRT Analytics Loader

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

www.actuate.com

Document No. 150331-2-580321 April 13, 2015

Contents

About *Using BIRT Analytics Loader* v

Chapter 1
Introducing BIRT Analytics Loader 1
About BIRT Analytics Loader 2
Understanding BIRT Analytics Loader 2
Understanding BIRT Analytics users 3
Getting started 3

Chapter 2
Working with projects 5
About BIRT Analytics Loader projects 6
Planning a project 6
Understanding the FastDB repository 7
Creating a new project 7
Managing projects 9
Laying out the feature tabs 10
Running a load process 10
Stopping the Tomcat service 10
Stopping using Services 10
Running a project 11
Using reports 12
Synchronizing data 12
Synchronizing data using BIRT Analytics Admin 12
Synchronizing data using the Postload function 16
Managing FastDB Engine accounts 16

Chapter 3
Extracting data 19
About extracting data 20
Connecting to data 20
Connecting to databases 20
Sharing databases 21
Testing a Datasource connection 21
Using native drivers 22
Using ODBC drivers 24
Connecting to files 26
Uploading local files 27
Uploading remote files 27
Uploading files using a file list 28
Updating data source definitions 29
Creating a load process 29
About transactions 29
Setting load process options 30
Ignoring items during the load process 30
Creating a load transaction 30
Mapping data from files 31
Defining a target database 32
Defining a source file format 33
Defining table columns 34

Mapping data from databases 38
Defining a data set 39
How to manage the loading process when exceptions occur 41
ON ERROR Attribute 41

Chapter 4
Managing data 43
About managing data 44
Understanding Explorer 44
Refreshing the repository content 45
Exploring table properties 45
Exploring column properties 45
Dropping database objects 47
Understanding dynamic and permanent columns 47
Understanding Links 48

Chapter 5
Transforming data 51
About transforming data 52
Ignoring items during the transform process 52
Understanding Includes 53
Understanding scripts 53
Understanding Repository 55
Using the text editor 55
Creating new columns 55
AGGREGATE 56
DECODE 57
DOMAIN 60
EXPRESSION 62
Supported formats when working with DATE, TIME or DATETIME 67
INUMERIC 69
PARAMETRIC 71
QUANTILE 74
RANKING 75
Managing data objects 77
DROPCOLUMN 77
DROPDATABASE 78
DROPTABLE 78
DROPLINK 78
LINK 79
MAKEPERMANENT 80
RENAME 81
Miscellaneous instructions 81
EXPORT 82
IFEXISTS 85
IFLINKED 86
INCLUDE 87
PRINT 88
VAR 88
How to manage the transformation process when exceptions occur 90
ON ERROR Instructions 90

Chapter 6
Administering BIRT Loader 91

About BIRT Loader administration 92
Managing users 92
Scheduling load processes 93
Using External call 95
Running projects using qLauncher 96
Understanding the BIRT Loader internal database 98
Copying a load project to a new location 99
Managing the repository 100
Creating a new database repository 101
Copying a data repository to a new location 102
Changing the repository content using the same location 103
Adding a BIRT Analytics database to a repository 103

Glossary 105

Index 119

About Using BIRT Analytics Loader

BIRT Analytics is a web application, data repository, and set of web services that support analysis and forecasting of big data.

BIRT Analytics consists of three key components:

- Actuate BIRT Analytics, a tool that is used to carry out dynamic analyses
- BIRT Analytics Administration, a set of tools that supports administering access and privileges for each module and component of the BIRT Analytics product
- 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 Actuate BIRT 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. Extracting data.* This chapter describes how to create data connections and data mappings.
- *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.
- *Chapter 6. Administering BIRT 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.

Chapter

1

Introducing BIRT Analytics Loader

This chapter contains:

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

About BIRT Analytics Loader

Business analytics depend on sufficient volumes of high-quality data. The difficulty in ensuring data quality is integrating and reconciling data across different systems, and then deciding what subsets of data to make available to the analysis tool.

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

- Extract reads data from a database.
- Transform converts the extracted data from its previous form into the form required for placement in another database. The transformation process uses rules, or lookup tables, or combines the data with other data.
- Load writes the data into the target database.

Use BIRT Analytics Loader to load big data for analysis. Use BIRT Analytics to analyze the data. Both tools run in a web browser, and ship as options in the Actuate BIRT Analytics product.

Understanding BIRT Analytics Loader

BIRT Analytics Loader is a web-based application you use to create and execute processes that build databases for data mining and reporting. Using BIRT Loader, you can schedule the load processes to run at convenient times. As the diagram in Figure 1-1 shows, BIRT Analytics Loader is installed on a Tomcat web server.

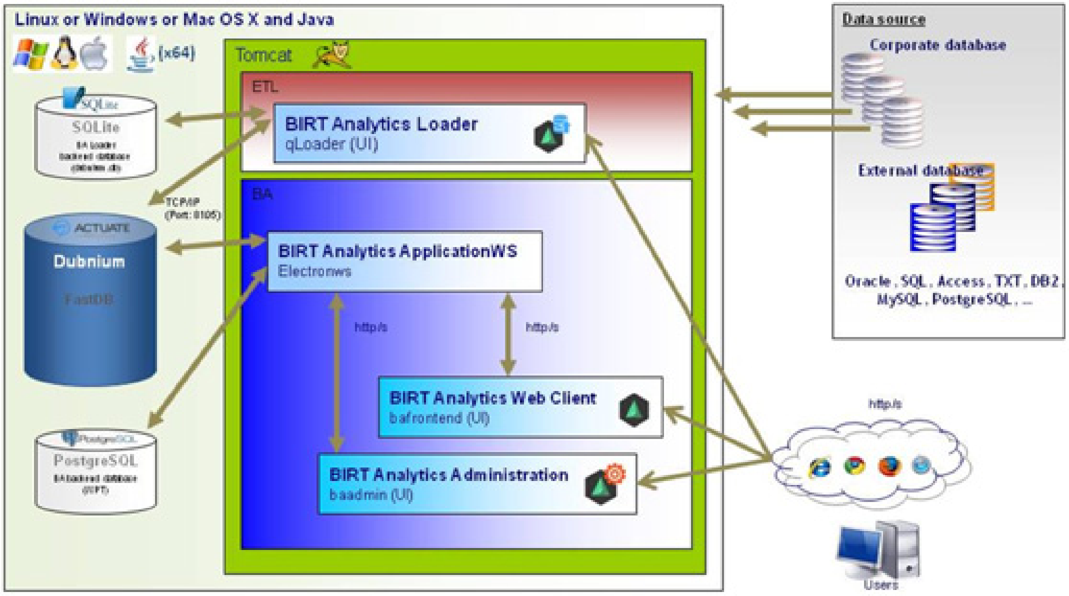


Figure 1-1 Actuate BIRT Analytics architecture

BIRT Analytics Loader supports connections to different data sources, relational databases, and text files, using database native drivers and ODBC connectors. The application builds processes that extract, transform, and load data, storing it in a proprietary format in a FastDB database. Actuate FastDB database is a file system database, managed by the FastDB engine. The engine runs as a service, named Dubnium Service. You can create multiple databases and select which one to be active.

Use BIRT Analytics to analyze and report from this data. Use BIRT Analytics Administration tool to synchronize the data, imported by BIRT Loader, with BIRT Analytics.

Understanding BIRT Analytics users

There are four types of users in Actuate BIRT Analytics products:

- BIRT Analytics users
  Users who log in to BIRT Analytics to do data mining and analysis.
- BIRT Analytics Administration users
  Users who log in to BIRT Analytics Administration and 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 projects for loading external data into a FastDB analytical database.
- FastDB Engine users
  The FastDB Engine user is the internal account the Dubnium server uses to communicate with the Actuate 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 Dubnium server data security account, see "Managing FastDB Engine accounts" in Chapter 6, "Administering BIRT 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:8080/qloader

   where

   - <webserver>:8080 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 displays the BIRT Analytics Loader login page.

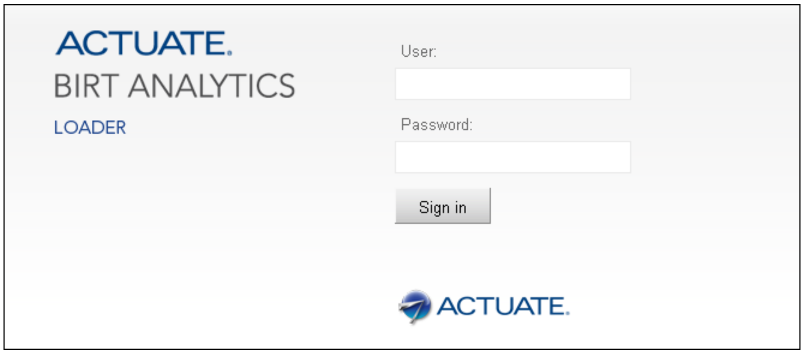


Figure 1-2 Logging in to Actuate BIRT Analytics

2 To log in, provide the following login credentials:

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

By default, BIRT Analytics Loader provides an administrator user account, "Administrator", with the password, "PASSWORD". This user is stored into the dubnium.db sqlite database. The user can be modified by editing this database. Although "Administrator" is the User name used for logging on to all the UI elements: BAfrontend, BAadmin and QLoader, the User name required for logging into Dubnium is "SYSTEM".

For more information about dubnium.db sqlite database, see "Understanding the BIRT Loader internal database" in Chapter 6, "Administering BIRT Loader."

3 Choose Sign in. The BIRT Analytics Loader home page associated with your login credentials appears, as shown in Figure 1-3.

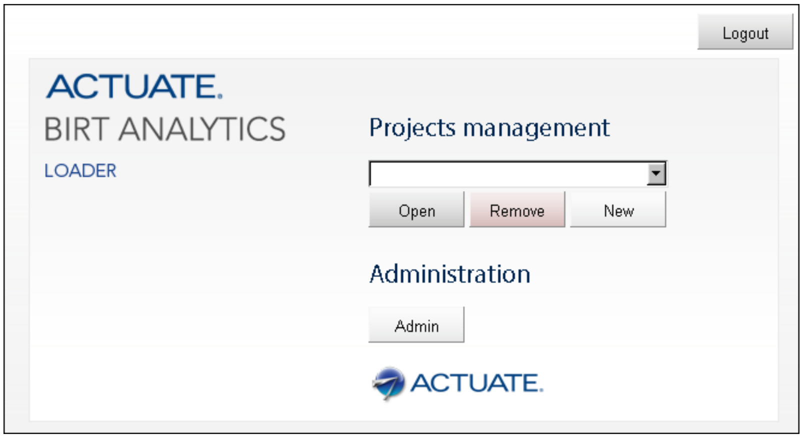


Figure 1-3 Viewing the BIRT Analytics Loader home page

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 one.

An administrator user, such as "Administrator", sees the Administrator tab. Options to manage user privileges and schedule loading processes appear on Administrator.

How to log out of Actuate BIRT Analytics

Choose Logout to quit a user session of BIRT Analytics Loader.

When working on a project, you can log out by choosing the logout button on the top right corner of the page.

Chapter

2

Working with projects

This chapter contains:

- About BIRT Analytics Loader projects
- Planning a project
- 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 you will extract, load, and transform the data into the target database. A BIRT Loader non-administrator user works with only the projects he creates. 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

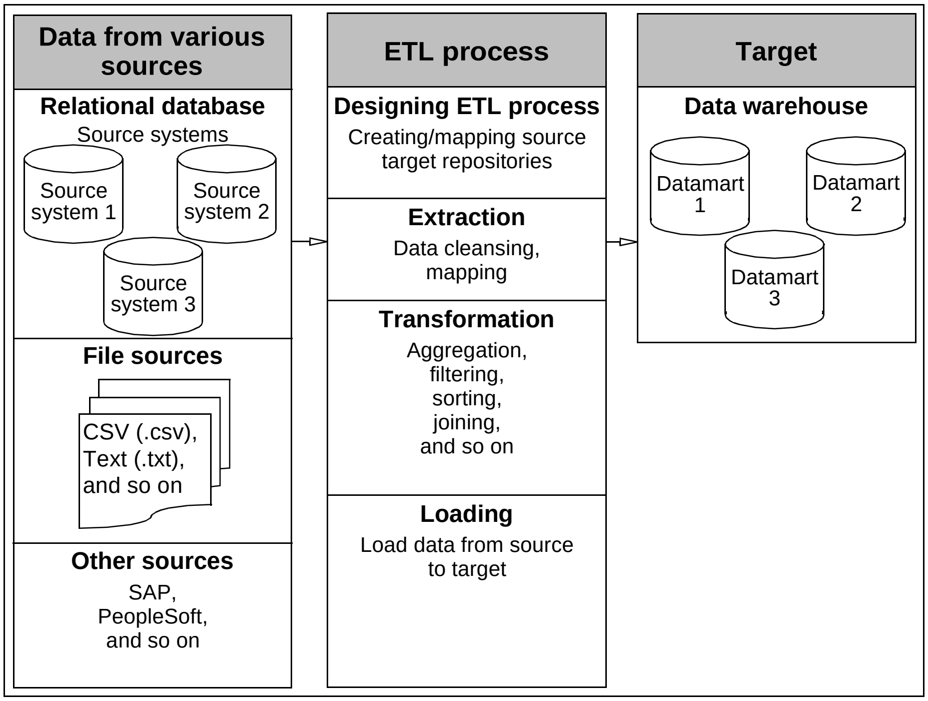


Figure 2-1 Sample ETL process

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 be extracting, cleansing, and transforming your data sources to get 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 can be as simple as a data refresh, cleaning old data, and loading new data. In some occasions, the update could require appending data based on complex criteria such as keyed reads or joins against an

existing dimension before writing the new or updated rows to a database table. Building these rules may 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 the data will be transformed, if at all

Understanding the FastDB repository

BIRT Analytics Loader loads data into a FastDB database repository, operated by the FastDB Engine. When FastDB is installed, a default 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 allows you to create a different location for your database repository and set up BIRT 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 Loader.”

Creating a new project

When you log in to BIRT Analytics Loader for the first time, there are no existing projects, and the Open and Remove tabs are grayed out, as shown in Figure 2-2.

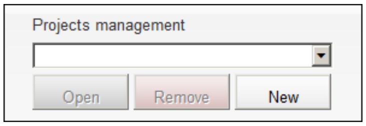


Figure 2-2 Creating a new project

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

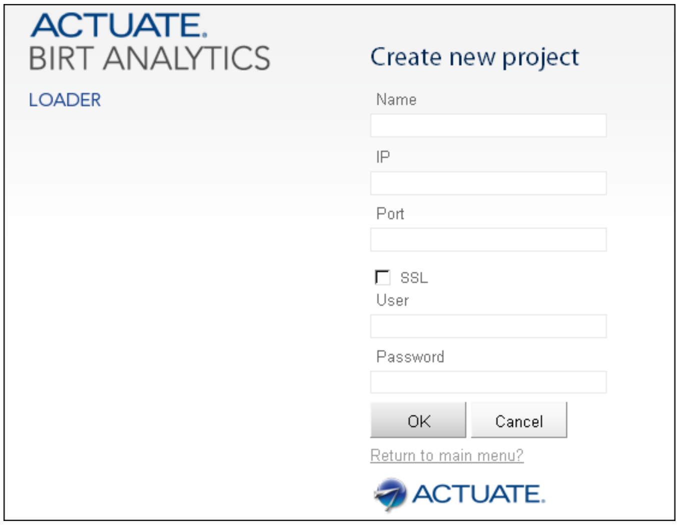


Figure 2-3 Setting a new project

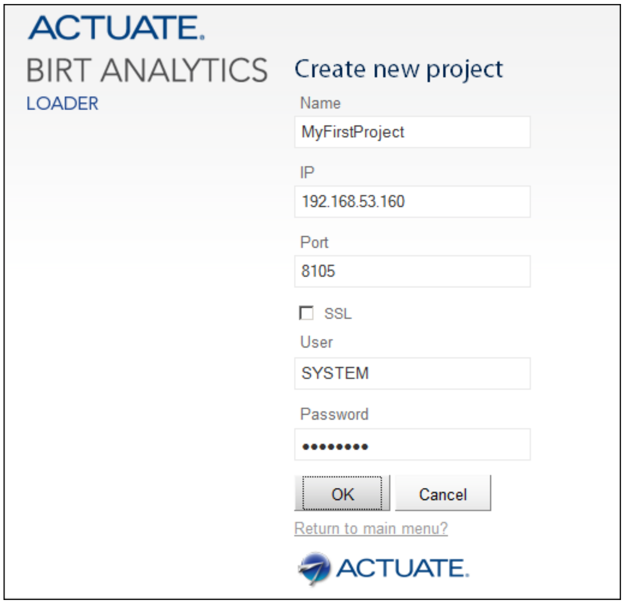
BIRT Loader users can access only projects they create. A BIRT Loader administrator can also only access his/her own projects - except in the case of other users that have previously shared a project with him/her.

How to create a new project

1 To create a new project, you set the following parameters:

- In Name, type the name of the project.
- In IP, type the IP address of the web server where BIRT Analytics Loader is installed.
- In Port, type the port number that the BIRT Analytics Loader application uses. The default port number is 8105.
- Choose SSL if a secure connection is used. In this case the default port number is 8106.
- In User, type the FastDB Engine user. The default user is SYSTEM.
- In Password, type the FastDB Engine user's password. The default password is PASSWORD.

Figure 2-4 displays the settings of a project named MyFirstProject.



Create new project

Name
MyFirstProject

IP
192.168.53.160

Port
8105

☐ SSL

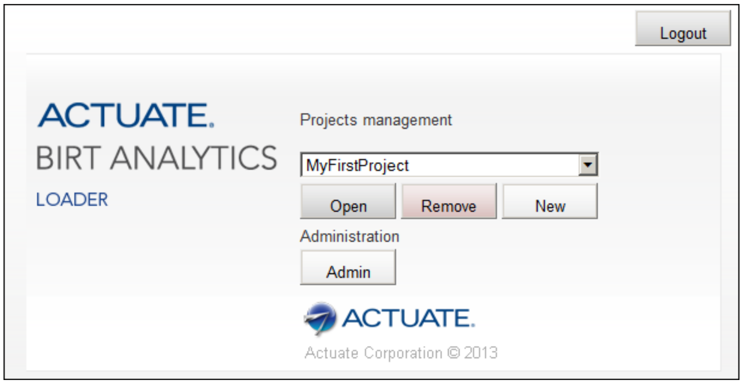
User
SYSTEM

Password
••••••••

o

Figure 2-4 Setting up MyFirstProject

2 Choose OK to create the project. The Projects management page appears and the project name is in the drop-down list, as shown in Figure 2-5.



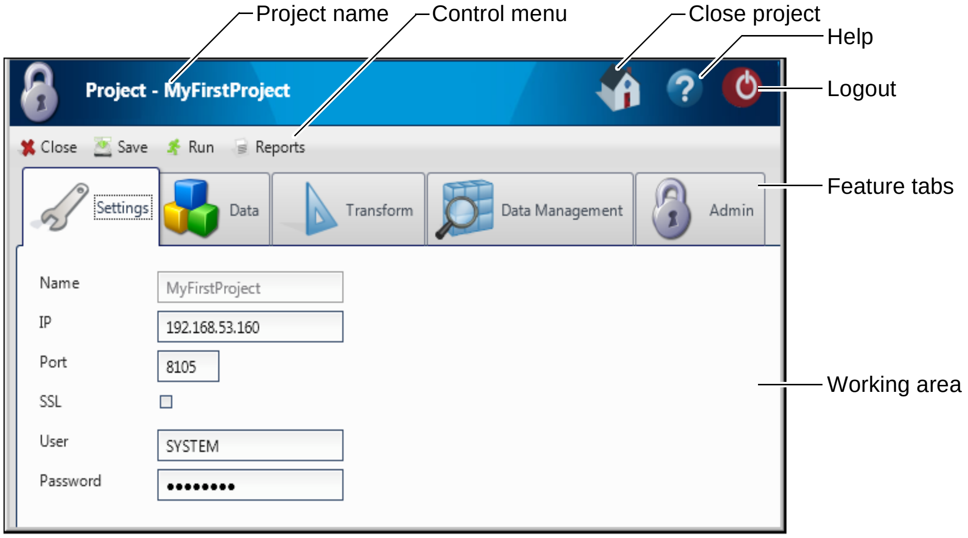
Projects management

MyFirstProject

Figure 2-5 Opening an existing project

Managing projects

After opening a project using BIRT Analytics Loader, the project development page appears, as shown in Figure 2-6. The project page contains a top banner, a control menu, feature tabs, and work area. The top banner displays the project name and the logout link.



Name MyFirstProject
IP 192.168.53.160
Port 8105
SSL □
User SYSTEM
Password ••••••••

Figure 2-6 Developing a project

The control menu contains the links shown in Table 2-1 for managing a project.

Table 2-1 Icons for the control menu

Icon	Label	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.
	Reports	Shows the load process log files.

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. The project name cannot be changed.
	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.
	Data Management	Tools used to browse and explore data in the target database, and create and remove links between data columns in tables.

(continues)

Table 2-2 Icons for the feature tabs (continued)

Icon	Label	Purpose
🔒	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, it is recommended that every user logs out before closing the browser.

Laying out the feature tabs

A tools menu provides customization options for the feature tabs in the user interface, 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.

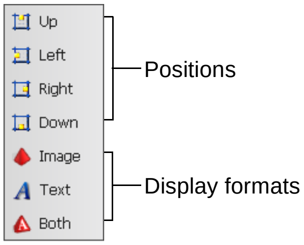


Figure 2-7 Options for tool tab placement and text

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 offline. For more information on how to schedule a load process, check “Scheduling load processes” in Chapter 6, “Administering BIRT Loader.”

Stopping the Tomcat service

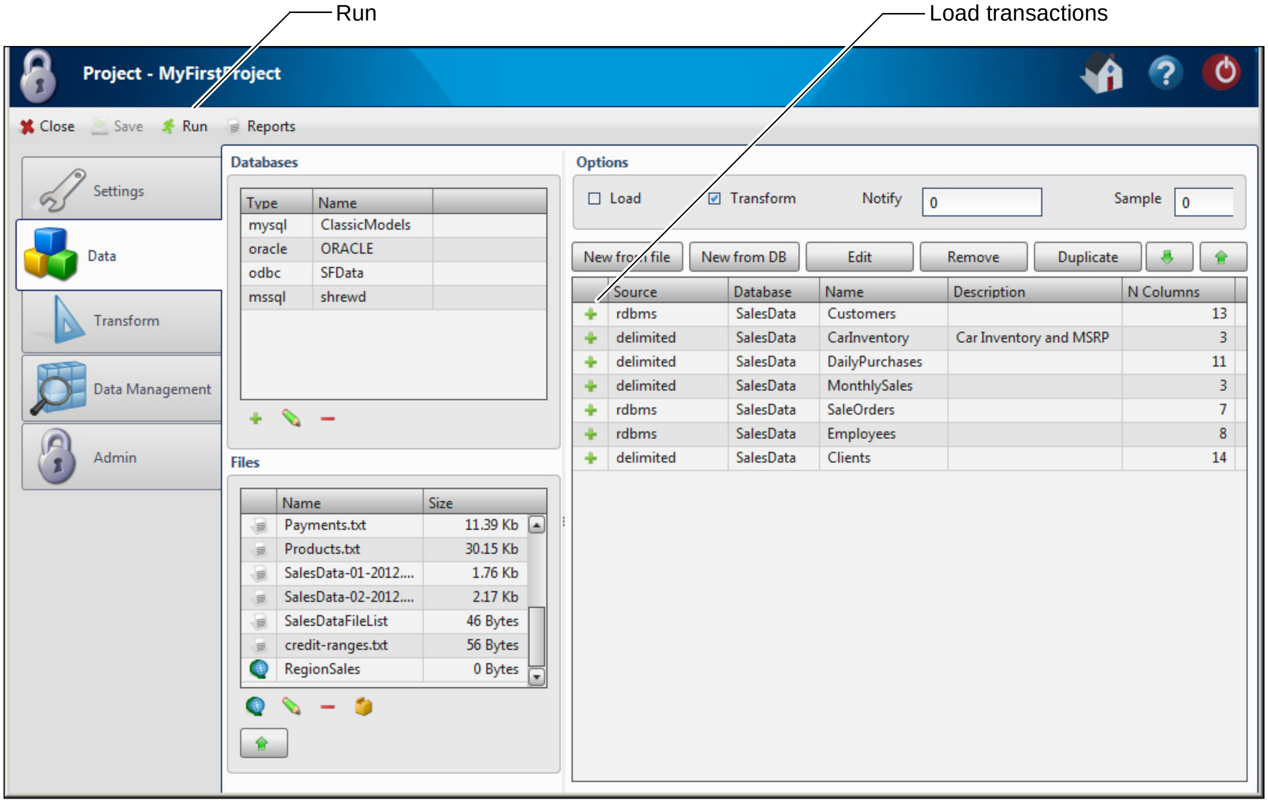
BIRT Analytics installs on Tomcat Apache. To prevent end users from logging in to BIRT Analytics, stop the web server on which BIRT Analytics runs, using either a command-line editor or Windows Services.

Stopping using Services

You use Services to stop, start, or restart a service.

Running a project

To run a loading project, you must choose Run, as shown in Figure 2-8.



Databases

Type	Name
mysql	ClassicModels
oracle	ORACLE
odbc	SFData
mssql	shrewd

Files

Name	Size
Payments.txt	11.39 Kb
Products.txt	30.15 Kb
SalesData-01-2012....	1.76 Kb
SalesData-02-2012....	2.17 Kb
SalesDataFileList	46 Bytes
credit-ranges.txt	56 Bytes
RegionSales	0 Bytes

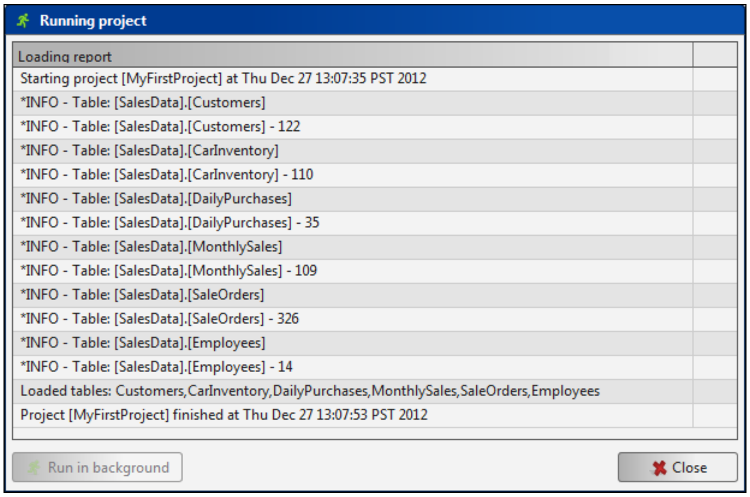
Options

☐ Load ☑ Transform Notify 0 Sample 0

Source	Database	Name	Description	N Columns
rdbms	SalesData	Customers		13
delimited	SalesData	CarInventory	Car Inventory and MSRP	3
delimited	SalesData	DailyPurchases		11
delimited	SalesData	MonthlySales		3
rdbms	SalesData	SaleOrders		7
rdbms	SalesData	Employees		8
delimited	SalesData	Clients		14

Figure 2-8 Running MyFirstProject

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



```
Starting project [MyFirstProject] at Thu Dec 27 13:07:35 PST 2012
*INFO - Table: [SalesData].[Customers]
*INFO - Table: [SalesData].[Customers] - 122
*INFO - Table: [SalesData].[CarInventory]
*INFO - Table: [SalesData].[CarInventory] - 110
*INFO - Table: [SalesData].[DailyPurchases]
*INFO - Table: [SalesData].[DailyPurchases] - 35
*INFO - Table: [SalesData].[MonthlySales]
*INFO - Table: [SalesData].[MonthlySales] - 109
*INFO - Table: [SalesData].[SaleOrders]
*INFO - Table: [SalesData].[SaleOrders] - 326
*INFO - Table: [SalesData].[Employees]
*INFO - Table: [SalesData].[Employees] - 14
Loaded tables: Customers,CarInventory,DailyPurchases,MonthlySales,SaleOrders,Employees
Project [MyFirstProject] finished at Thu Dec 27 13:07:53 PST 2012
```

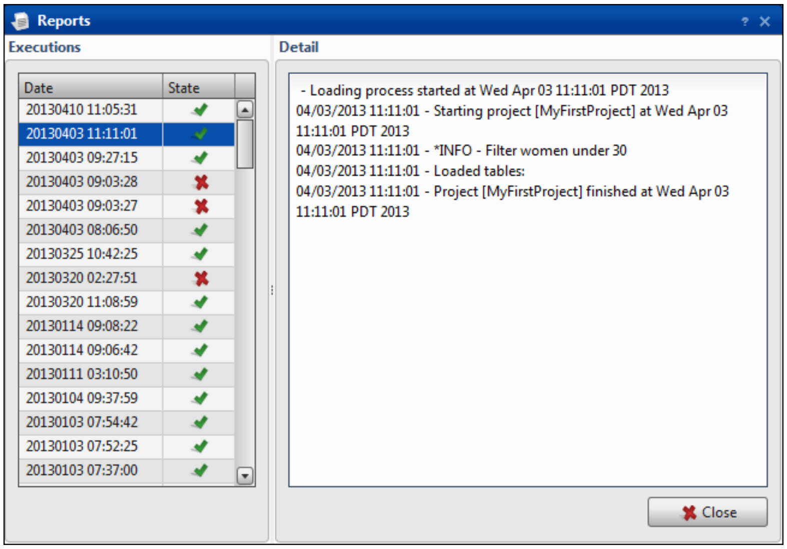
Figure 2-9 Running project log

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, set a number of records for which a log entry is created, in Options—Notify. For loading processes requiring a long time, choose Run in background, to minimize the Running project window. You can 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 appear in Executions, as shown in Figure 2-10. Date displays a date and time when a project is executed. State displays the generation status. After selecting a date, the log content appears in Detail.



Date	State
20130410 11:05:31	✓
20130403 11:11:01	✓
20130403 09:27:15	✓
20130403 09:03:28	✗
20130403 09:03:27	✗
20130403 08:06:50	✓
20130325 10:42:25	✓
20130320 02:27:51	✗
20130320 11:08:59	✓
20130114 09:08:22	✓
20130114 09:06:42	✓
20130111 03:10:50	✓
20130104 09:37:59	✓
20130103 07:54:42	✓
20130103 07:52:25	✓
20130103 07:37:00	✓

```
 - Loading process started at Wed Apr 03 11:11:01 PDT 2013
04/03/2013 11:11:01 - Starting project [MyFirstProject] at Wed Apr 03
11:11:01 PDT 2013
04/03/2013 11:11:01 - *INFO - Filter women under 30
04/03/2013 11:11:01 - Loaded tables:
04/03/2013 11:11:01 - Project [MyFirstProject] finished at Wed Apr 03
11:11:01 PDT 2013
```

Figure 2-10 Browsing reports

Synchronizing data

Synchronizing data is the final step in the loading process. In BIRT Analytics, you can synchronize data two ways. You can use the BIRT Analytics Admin tool, or the Post Loading process.

Synchronizing data using BIRT Analytics Admin

The Synchronize option in BIRT Admin allows the administrator to update the application database to contain the most recent changes in the BIRT Analytics Engine repository. Execute this option 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. Synchronizing is also required when you change the data, after increasing, reducing, or modifying the records in a database.

After synchronizing the data you must set the privileges for the users or groups to provide access to the database. In the BIRT Analytics security options list, the ACL to objects option allows the administrator to create, modify, or delete a security group or access control list (ACL) and manage privileges over database objects.

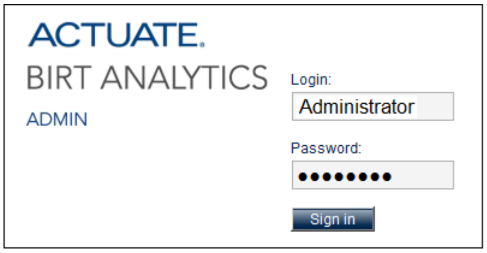
How to synchronize data using BIRT Analytics Admin

To do the synchronization you must have access to the BIRT Analytics Admin application.

1 Using your web browser, connect to the following site:

```
http://localhost/qadmin
```

The BIRT Analytics login screen appears as shown in Figure 2-11.



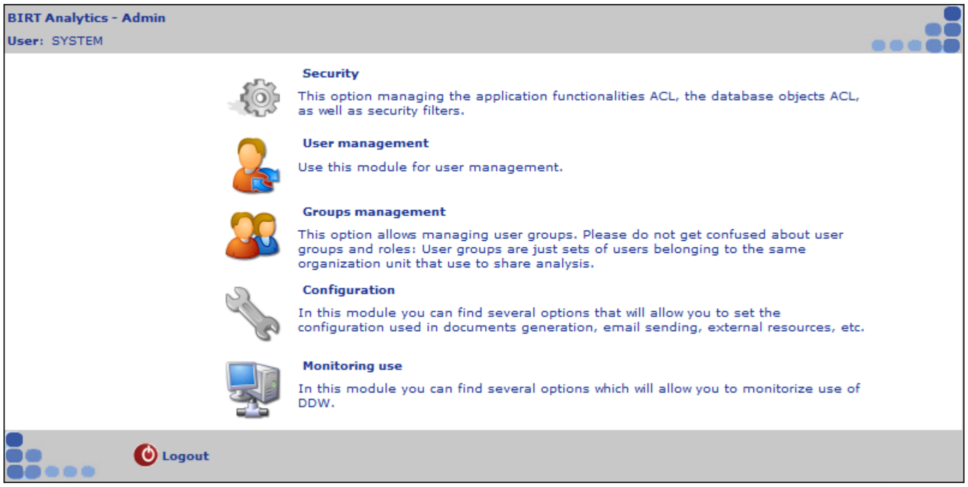
ACTUATE.
BIRT ANALYTICS
ADMIN

Login: Administrator

Password: ••••••••

Figure 2-11 Logging in to BIRT Analytics Admin

2 In Login, type Administrator. In Password, type PASSWORD. Choose Sign in. The BIRT Analytics Admin page appears, as shown in Figure 2-12.



BIRT Analytics - Admin
User: SYSTEM

Security
This option managing the application functionalities ACL, the database objects ACL, as well as security filters.

User management
Use this module for user management.

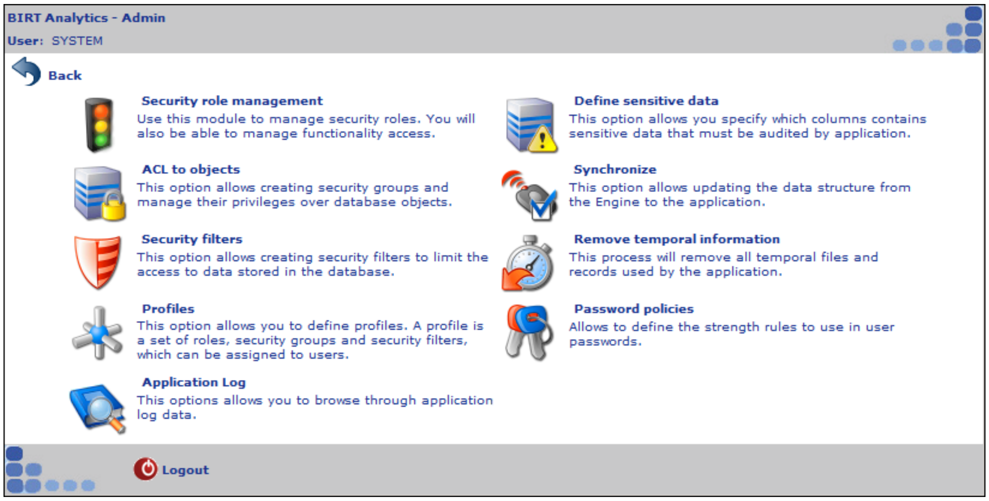
Groups management
This option allows managing user groups. Please do not get confused about user groups and roles: User groups are just sets of users belonging to the same organization unit that use to share analysis.

Configuration
In this module you can find several options that will allow you to set the configuration used in documents generation, email sending, external resources, etc.

Monitoring use
In this module you can find several options which will allow you to monitorize use of DDW.

Figure 2-12 Accessing the security administration

3 In BIRT Analytics Admin, choose Security. The Security page appears, as shown in Figure 2-13.



BIRT Analytics - Admin
User: SYSTEM

Security role management
Use this module to manage security roles. You will also be able to manage functionality access.

ACL to objects
This option allows creating security groups and manage their privileges over database objects.

Security filters
This option allows creating security filters to limit the access to data stored in the database.

Profiles
This option allows you to define profiles. A profile is a set of roles, security groups and security filters, which can be assigned to users.

Application Log
This options allows you to browse through application log data.

Define sensitive data
This option allows you specify which columns contains sensitive data that must be audited by application.

Synchronize
This option allows updating the data structure from the Engine to the application.

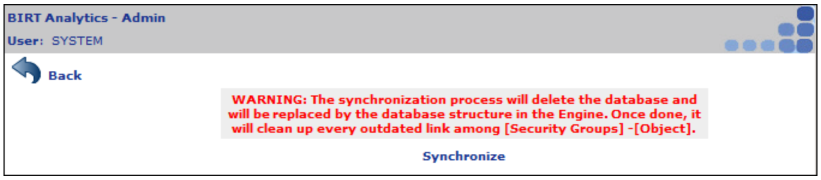
Remove temporal information
This process will remove all temporal files and records used by the application.

Password policies
Allows to define the strength rules to use in user passwords.

Figure 2-13 Accessing the synchronize settings

4 In Security, choose Synchronize.

5 In Synchronize, choose Synchronize again, as shown in Figure 2-14.



WARNING: The synchronization process will delete the database and will be replaced by the database structure in the Engine. Once done, it will clean up every outdated link among [Security Groups] -[Object].

Figure 2-14 Updating the data structure from the engine

6 Choose Back when the synchronization has finished.

7 To make a synchronized database visible to a user security group, complete the following steps:

1 In BIRT Analytics Admin, choose ACL to objects.

2 Select a security group for which to assign access privileges. Then, choose Modify, as shown in Figure 2-15.

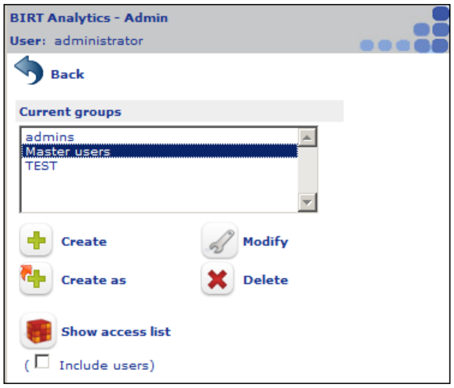
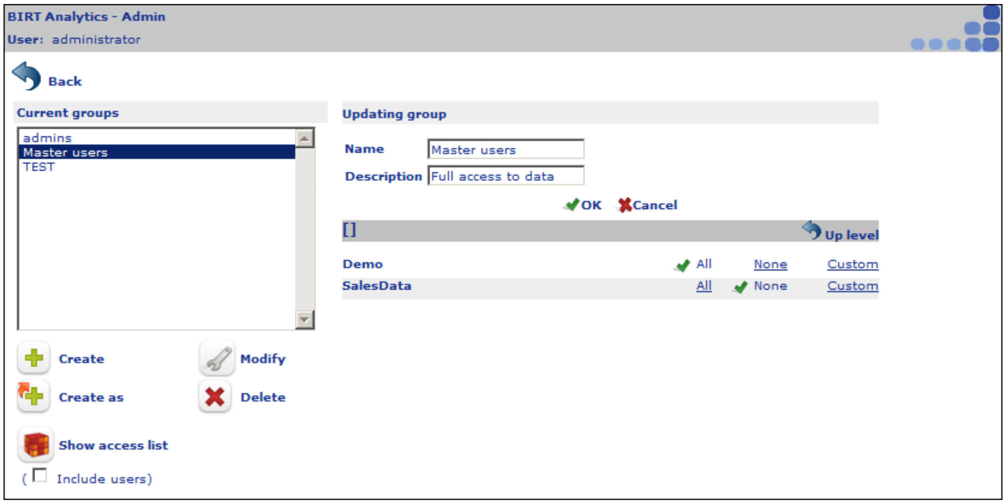


Figure 2-15 Selecting a user security group

A list of available databases appears, as shown in Figure 2-16.



Name: Master users

Description: Full access to data

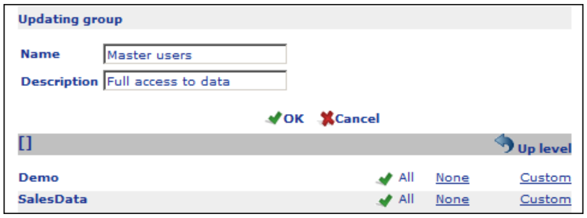
Demo	✓ All	None	Custom
SalesData	All	✓ None	Custom

Figure 2-16 Granting a user group access privilege to a database

3 Select a database to which you will grant a user group access privileges. For example, select the SalesData database.

4 To grant a group full access to a database, choose All for that database. The modifications appear in Updating group. For example, Figure 2-17 shows a group

called Master users, modified to have all privileges for, or full access to, the SalesData database.


Updating group

Name: Master users

Description: Full access to data

Demo	✓ All	None	Custom
SalesData	✓ All	None	Custom

Figure 2-17 Updating group

5 Choose OK.

6 Choose Back.

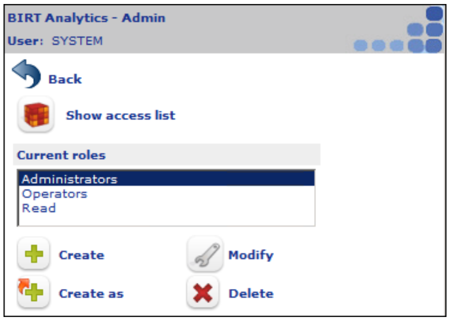
How to verify functionality activation

To verify that all functionalities are activated for a user group, complete the following steps.

1 Choose a security role, as shown in Figure 2-18.

2 Select a user group, for example Administrators

3 Choose Modify.


BIRT Analytics - Admin

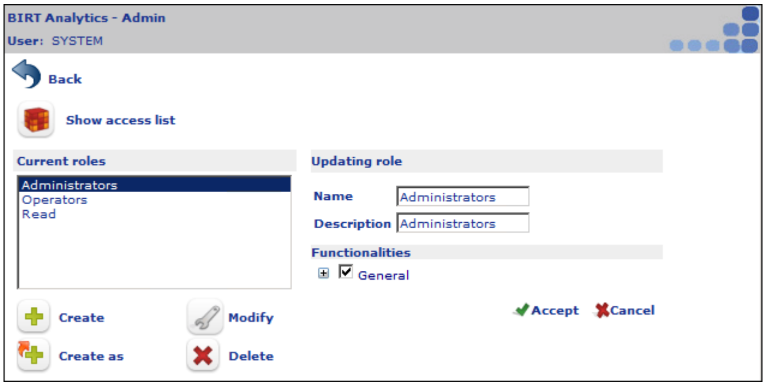
User: SYSTEM

Current roles

Administrators
Operators
Read

Figure 2-18 Selecting the Administrators role

4 In Functionalities, de-select General. Then, select General, as shown in Figure 2-19.


BIRT Analytics - Admin

User: SYSTEM

Current roles

Administrators
Operators
Read

Updating role

Name: Administrators

Description: Administrators

Functionalities

⊞ ☑ General

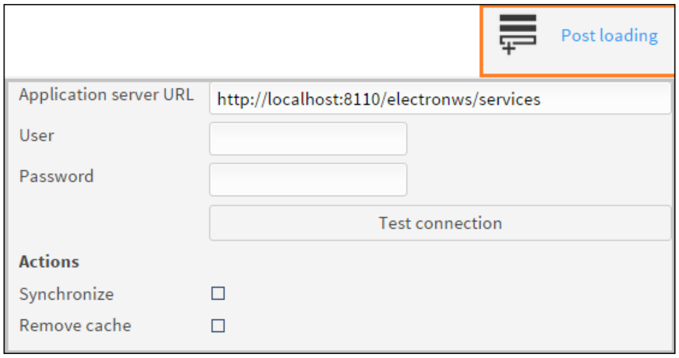
Figure 2-19 Selecting all functionalities

5 Choose OK.

6 Choose Logout to exit BIRT Analytics Admin.

Synchronizing data using the Postload function

BIRT Analytics Loader provides a post loading tool located in the Post loading tab, shown in Figure 2-20. In this tab users can define the URl of the electron server and the requested actions. This functionality allows you to automate a synchronization process and run it in an unattended manner (synchronizing an analytical database after loading data into the FastDB database. It also allows you to clear the cache.



Application server URL http://localhost:8110/electronws/services

Figure 2-20 Post loading interface

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 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 allows the system administrator to control user access to sensitive data by changing passwords or deleting FastDB Engine user accounts.

To manage FastDB Engine users, you use BIRT Loader Admin, as shown in Figure 2-21.

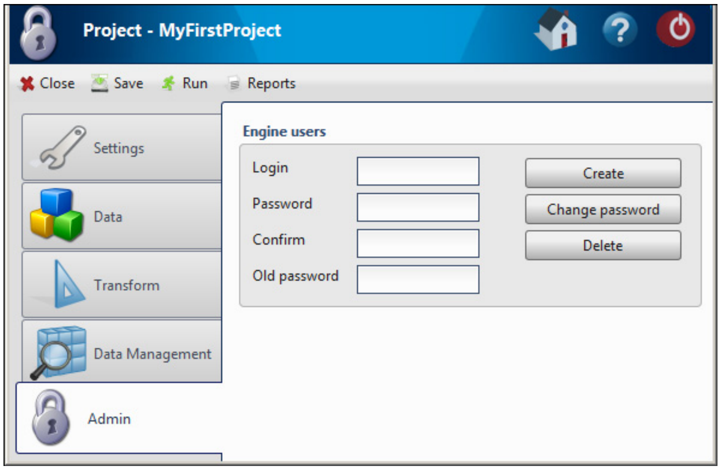


Figure 2-21 Managing FastDB Engine users

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 all of these fields:

- 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.

Chapter

3

Extracting data

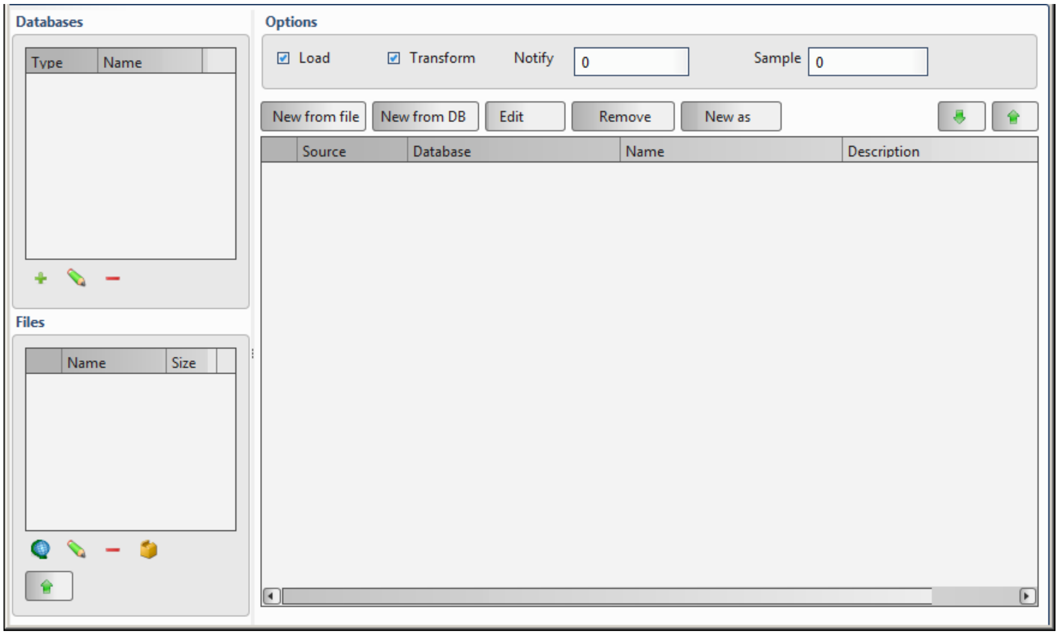
This chapter contains:

- About extracting data
- Connecting to data
- Creating a load process

About extracting data

The load process is composed of two main steps: loading and transformation. The configuration of the load process is done in the Data module.

The Data page, as shown in Figure 3-1, is divided into three areas: Databases, Files, and Options. Databases and Files are places where you define the connections to the data sources. Options is the place where you define a load process. A load process consists of options that determine the way the load process runs, and transactions. A load transaction describes the data mappings between the input data and the tables in the target database. A load process may contain multiple transactions.



☑ Load ☑ Transform Notify 0 Sample 0

Figure 3-1 Extracting data

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 as many data source connections as a load process requires. 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.

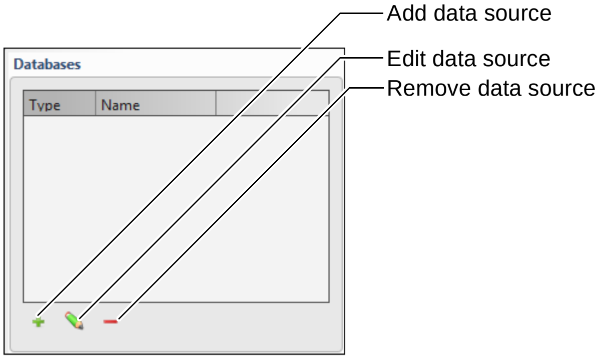
BIRT Analytics Loader connects to databases using native and ODBC drivers. BIRT Loader supports native drivers for the following database types:

- Microsoft SQL Server
- MySQL Database Server
- Oracle Database

- PostgreSQL

BIRT Analytics Loader also supports connecting to a database using an ODBC driver. To connect to a database having a native driver type not supported by BIRT Analytics Loader, you must provide an ODBC driver and set up an ODBC connection that uses the ODBC diver.

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

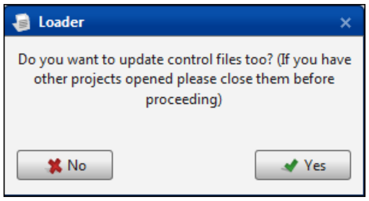


Add data source
Edit data source
Remove data source

Figure 3-2 Connection tools for database data sources

Updating data source definitions

You can modify the data source definitions by choosing the pencil icon. The changes in the data source definitions require an update of the related table definitions. Saving the changes displays a message, as shown in Figure 3-21, asking you to confirm the update.



Do you want to update control files too? (If you have other projects opened please close them before proceeding)

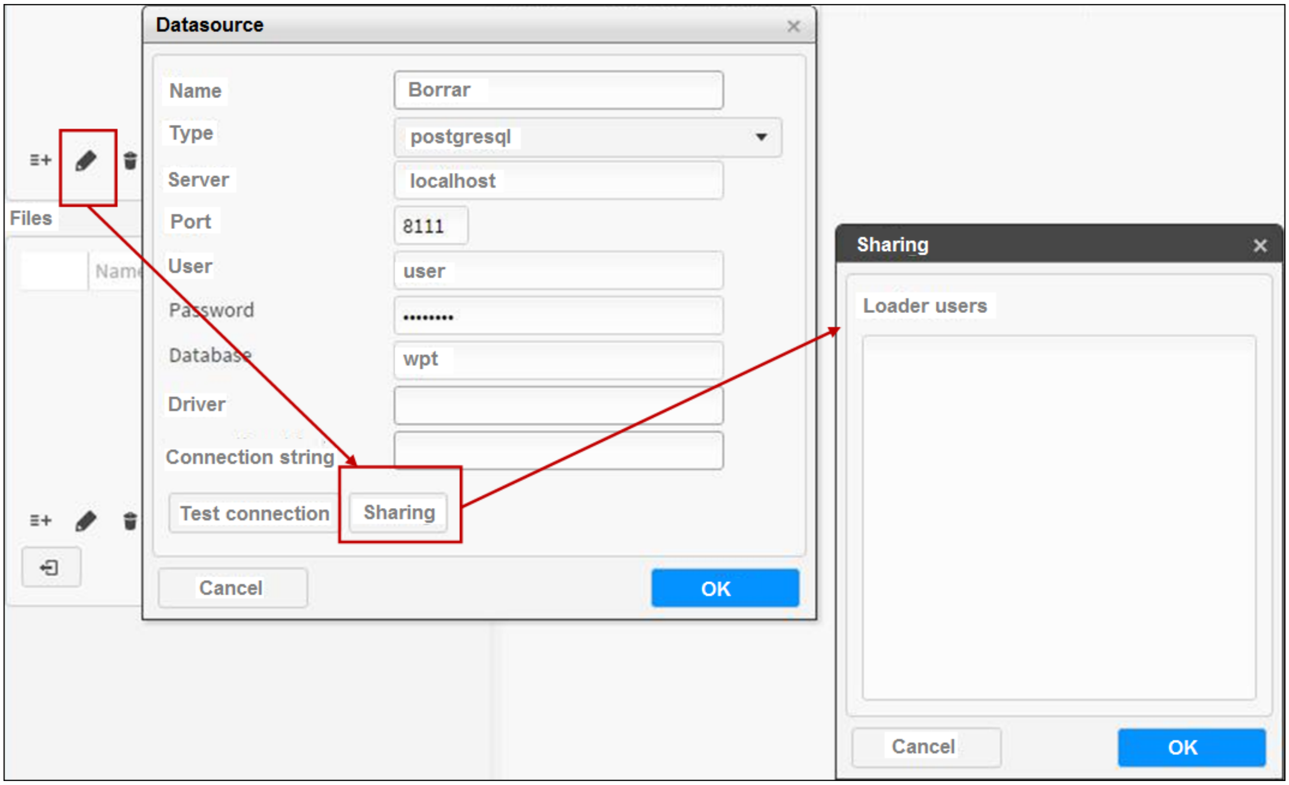
Figure 3-3 Updating data source message

Sharing databases

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

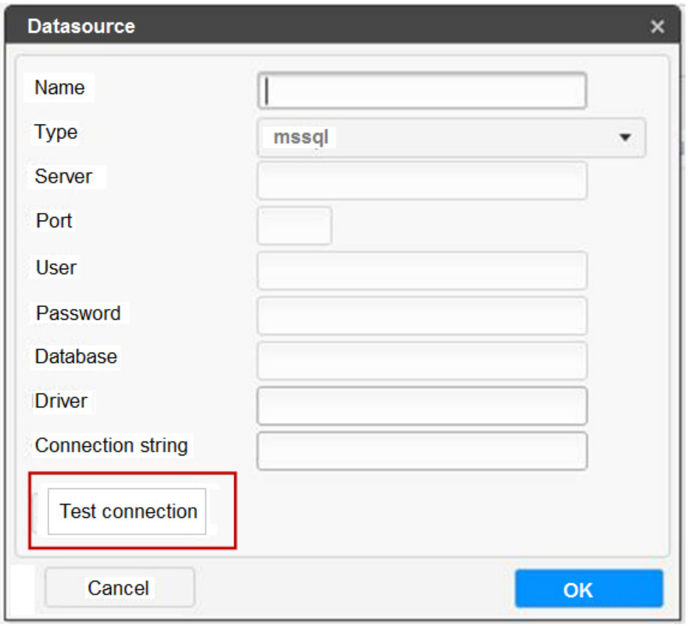
Testing a Datasource connection

It is possible to test the correctness of your connection to an external database (using native or ODBC drivers) when you are creating or editing an existing data source. This is done via the “Test Connection” button in the “Datasource” definition window. (See Figure 3-5 below).



Name: Borrar
Type: postgresql
Server: localhost
Port: 8111
User: user
Password: ••••••••
Database: wpt
Driver:
Connection string:

Figure 3-4 Sharing button in the Datasource definition screen

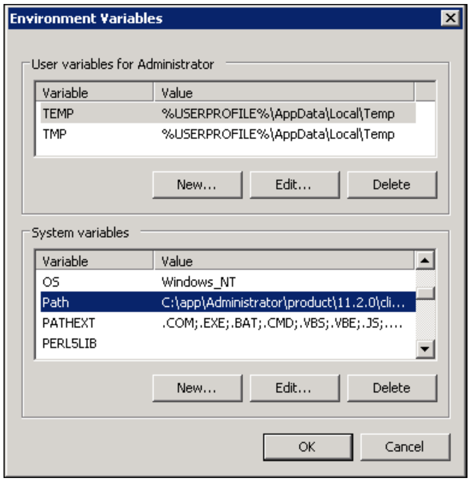


Name:
Type: mssql
Server:
Port:
User:
Password:
Database:
Driver:
Connection string:

Figure 3-5 The “Test connection” button in the “Datasource” definition window

Using native drivers

Before creating a connection using a native driver make sure the driver is installed on your machine. For MySQL server for example, check if the path to the driver is in the System Path. To see the System Path, use Control Panel➛System➛Advanced System Settings ➛Environment Variables➛Path, as shown in Figure 3-6.



Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable	Value
OS	Windows_NT
Path	C:\app\Administrator\product\11.2.0\cli...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PERL5LIB	

Figure 3-6 Editing Path variable

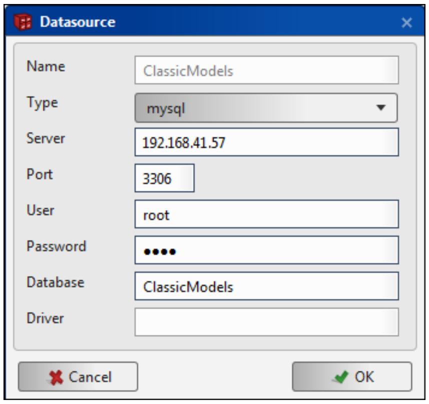
To create a data source connection you must provide the database connection information, as shown in Table 3-1. Contact your database administrator to obtain this information.

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

How to connect to a database using a native driver

1 In Databases, choose the + sign. Datasource appears, as shown in Figure 3-7.



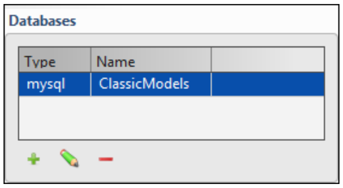
Name	ClassicModels
Type	mysql
Server	192.168.41.57
Port	3306
User	root
Password	••••
Database	ClassicModels
Driver	

Figure 3-7 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.

3 Choose OK to create the named data source connection. A new data source connection name and type appear in Databases, as shown in Figure 3-8.



Databases

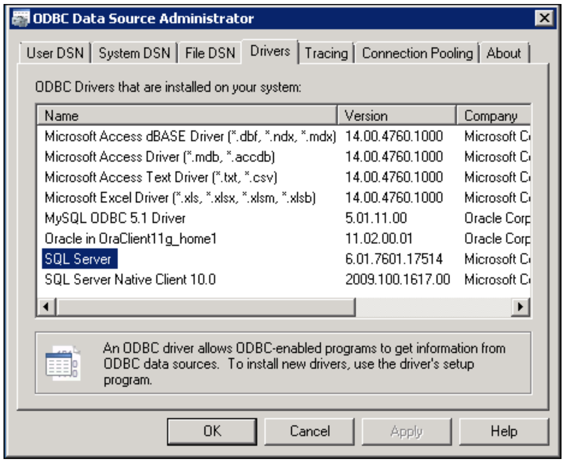
Type	Name
mysql	ClassicModels

Figure 3-8 A data source connection that uses 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 3-9.



ODBC Drivers that are installed on your system:

Name	Version	Company
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx)	14.00.4760.1000	Microsoft C
Microsoft Access Driver (*.mdb, *.accdb)	14.00.4760.1000	Microsoft C
Microsoft Access Text Driver (*.txt, *.csv)	14.00.4760.1000	Microsoft C
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)	14.00.4760.1000	Microsoft C
MySQL ODBC 5.1 Driver	5.01.11.00	Oracle Corp
Oracle in OraClient11g_home1	11.02.00.01	Oracle Corp
SQL Server	6.01.7601.17514	Microsoft C
SQL Server Native Client 10.0	2009.100.1617.00	Microsoft C

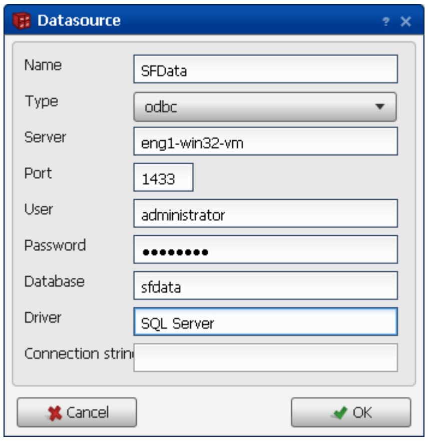
An ODBC driver allows ODBC-enabled programs to get information from ODBC data sources. To install new drivers, use the driver's setup program.

Figure 3-9 Finding an ODBC driver name

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 3-10:



Name: SFData

Type: odbc

Server: eng1-win32-vm

Port: 1433

User: administrator

Password: ••••••••

Database: sfdata

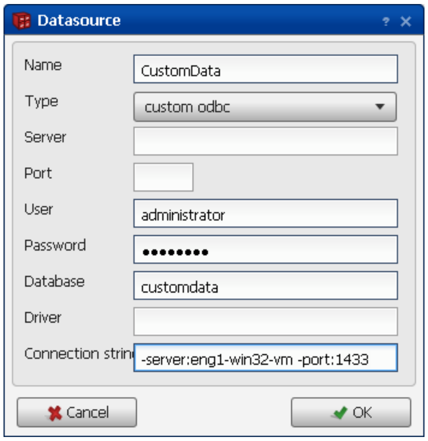
Driver: SQL Server

Connection strin

Figure 3-10 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, eng1-win32-vm, 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 3-11:



Name: CustomData

Type: custom odbc

Server:

Port:

User: administrator

Password: ••••••••

Database: customdata

Driver:

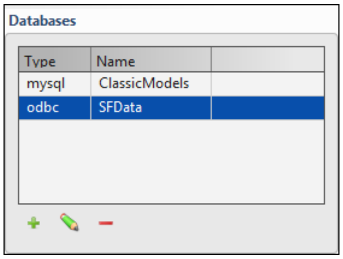
Connection strin: -server:eng1-win32-vm -port:1433

Figure 3-11 Using a 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 Databases, as shown in Figure 3-12.



Type	Name
mysql	ClassicModels
odbc	SFData

Figure 3-12 Database connections

Other options

Two additional options than can be used when connecting to databases are:

- *Sharing databases:* This option allows the sharing of database definition between users.
- *Test connection:* This option makes it possible to test the connection to an external database using native or ODBC drivers.

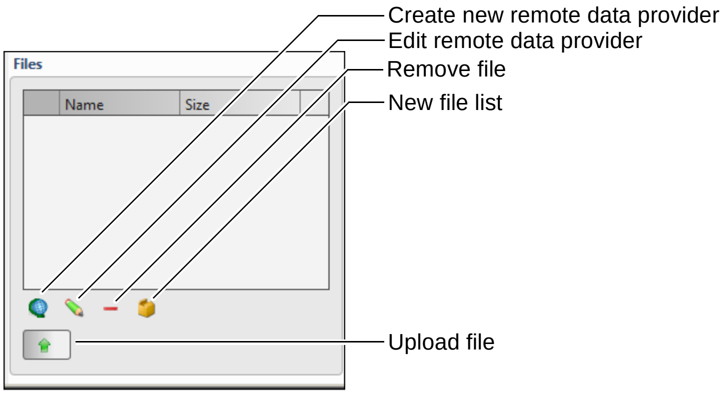
Connecting to files

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, make sure the file uses a valid structure. There is no limit to the size of text files. This is due to the fact that the *Default value* in *configuration.xml* is **-1**.

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

- Uploading a file from a local system
- Uploading a file from a remote system
- Uploading multiple files using a file list

Using BIRT Analytics Loader - Files, you manage data uploads from data files stored on local and remote systems, using the toolset shown in Figure 3-13.



Create new remote data provider
Edit remote data provider
Remove file
New file list
Upload file

Figure 3-13 Connection tools for flat file data sources

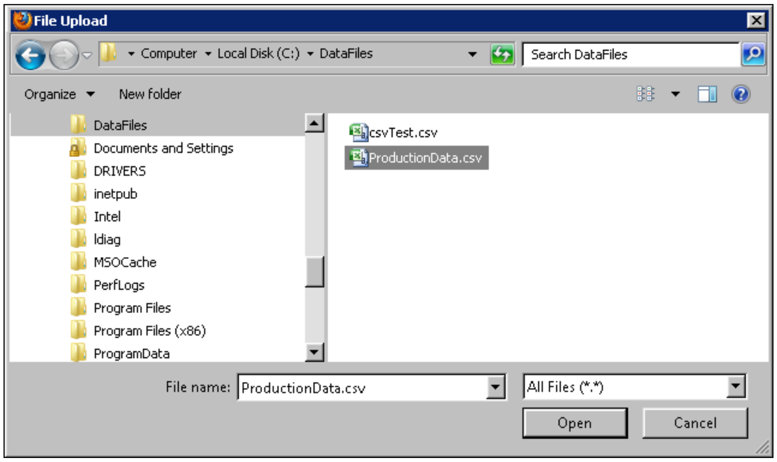
To prepare to access your text files you must know the file name and location.

Uploading local files

After selecting the file, it will be uploaded into the data folder of the load project. Because of network reasons there is a configurable maximum file size available.

How to upload a file from a local file system

1 In Files, choose the up arrow. File Upload appears, as shown in Figure 3-14.



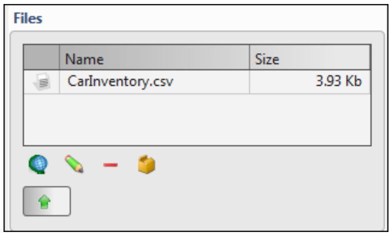
File name: ProductionData.csv

All Files (*.*)

Figure 3-14 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 Files, as shown in Figure 3-15. The local file is stored in the data folder of the project.



Name	Size
CarInventory.csv	3.93 Kb

Figure 3-15 Creating a local file data source

Uploading remote files

The remote data provider option allows you to load files using a web address. You have to choose a name, the Uniform Resource Identifier (URI), and the HTTP method (GET, POST).

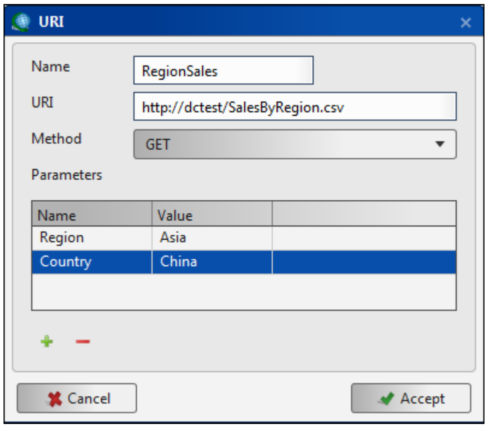
How to upload a file using URI

1 In Files, select the remote data provider icon. URI appears.

2 In URI, enter the following information, as shown in Figure 3-16:

- In Name, type the name you want to assign to this data source.
- In URI, type the URI, specifying the file location.
- In Method, select the method for obtaining the file. Contact the file provider to identify the method.
- In Parameters, enter parameters if needed.
    - Choose + to add parameter. In Name, type the parameter name. In Value, enter the parameter value, as shown in Figure 3-16.
    - To modify a parameter, select a column and change the entry.

- ❑ Choose - to remove a parameter.



Name: RegionSales

URI: http://dctest/SalesByRegion.csv

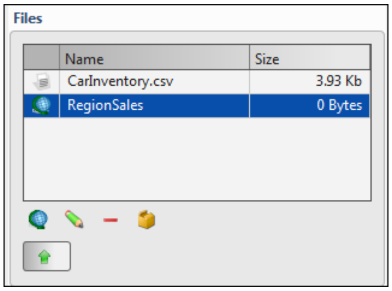
Method: GET

Parameters

Name	Value
Region	Asia
Country	China

Figure 3-16 Providing URI properties

3 Choose OK. The new flat file data source appears in Files, as shown in Figure 3-17.



Name	Size
CarInventory.csv	3.93 Kb
RegionSales	0 Bytes

Figure 3-17 Creating a remote file data source

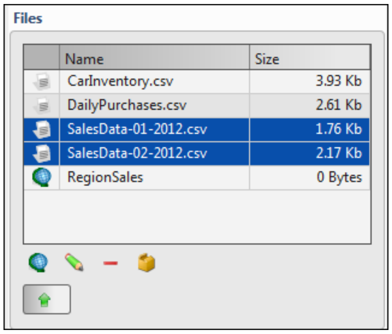
Uploading files using a file list

On many occasions, due to a large data volume and file size restrictions, 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 upload files using a file list

1 In Files, upload all the files you want to include in a list, the same way you upload local files.

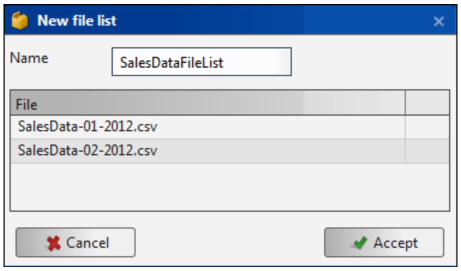
2 Press and hold Ctrl while using the mouse to select multiple file names, as shown in Figure 3-18.



Name	Size
CarInventory.csv	3.93 Kb
DailyPurchases.csv	2.61 Kb
SalesData-01-2012.csv	1.76 Kb
SalesData-02-2012.csv	2.17 Kb
RegionSales	0 Bytes

Figure 3-18 List file data source

3 In Files, choose New File list. File names that you selected in step 2 appear, as shown in Figure 3-19.



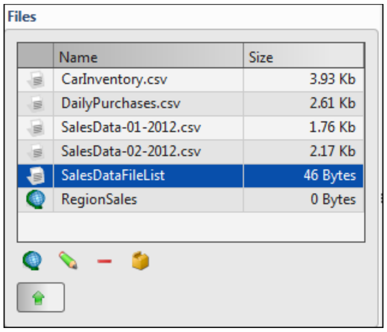
Name: SalesDataFileList

File
SalesData-01-2012.csv
SalesData-02-2012.csv

Figure 3-19 Selecting a list file

4 In Name, type a name for the new list file data source, then choose OK.

The list file data source appears in Files, as shown in Figure 3-20.

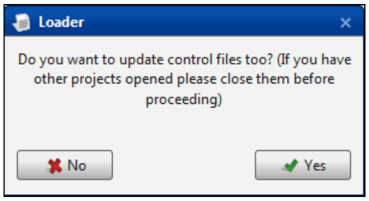


Name	Size
CarInventory.csv	3.93 Kb
DailyPurchases.csv	2.61 Kb
SalesData-01-2012.csv	1.76 Kb
SalesData-02-2012.csv	2.17 Kb
SalesDataFileList	46 Bytes
RegionSales	0 Bytes

Figure 3-20 Naming a list file data source

Updating data source definitions

You can modify the data source definitions by choosing the pencil icon. The changes in the data source definitions require an update of the related table definitions. Saving the changes displays a message, as shown in Figure 3-21, asking you to confirm the update.



Do you want to update control files too? (If you have other projects opened please close them before proceeding)

Figure 3-21 Updating data source message

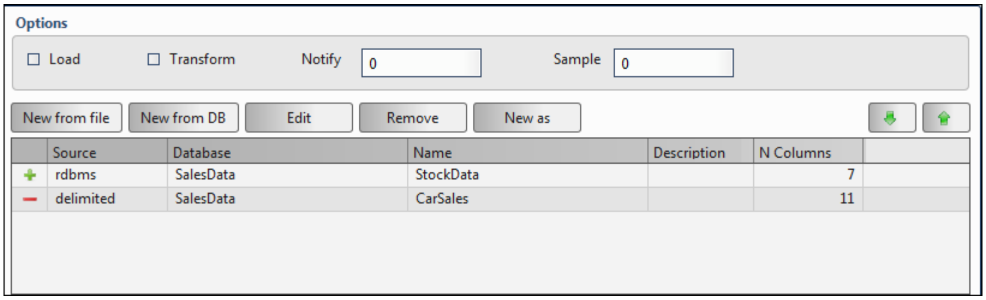
Creating a load process

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.

BIRT Analytics Options, for defining a process and tools for creating transactions, looks like the example shown in Figure 3-22.



Options

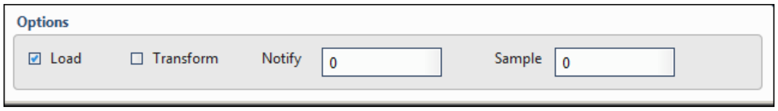
☐ Load ☐ Transform Notify 0 Sample 0

	Source	Database	Name	Description	N Columns
+	rdbms	SalesData	StockData		7
−	delimited	SalesData	CarSales		11

Figure 3-22 Defining data

Setting load process options

Use Options, shown in Figure 3-23, to select options that define a load process. Displayed settings that you save for each process execute when the process runs, as part of a data load project.



Options

☑ Load ☐ Transform Notify 0 Sample 0

Figure 3-23 Defining load process 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.

- Notify

  Type a value that sets a trigger to write a log message each time the set number of records loads.

- Sample

  Specify the number of records loaded for each table.

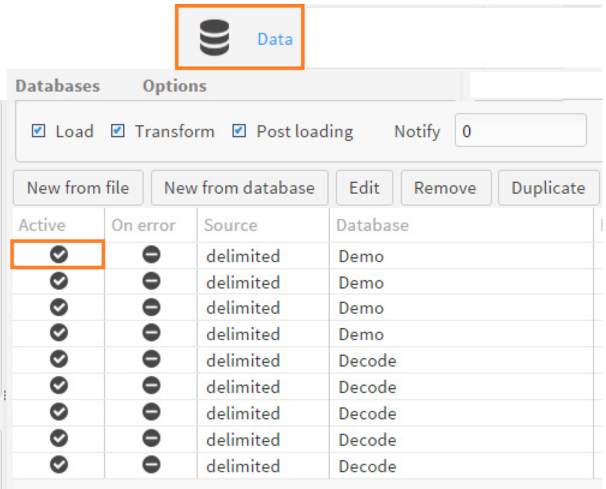
Ignoring items during the load process

Users can choose to have items (tables, instructions) ignored during the loading process in the Data tab (it can also be done in the Transform tab). Clicking on the “check” or “minus” symbols in the “Active” column in the Data tab, as shown in Figure 3-24, allows Users to have chosen elements ignored by 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 is the definition of the relationship and data flow between source and target objects.

Before building an actual data map for the loading process, you have to define a data model for target even if it is not an actual database. It will help you understand the entities to which it belongs. Then it will be 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.



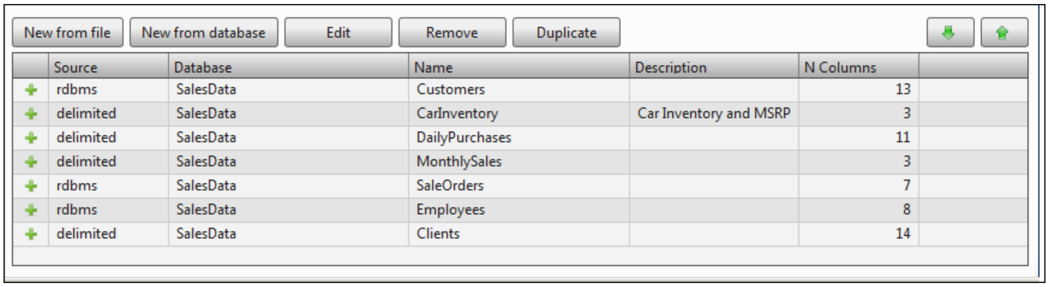
☑ Load ☑ Transform ☑ Post loading Notify 0

Active	On error	Source	Database
✓	⊖	delimited	Demo
✓	⊖	delimited	Demo
✓	⊖	delimited	Demo
✓	⊖	delimited	Demo
✓	⊖	delimited	Decode
✓	⊖	delimited	Decode
✓	⊖	delimited	Decode
✓	⊖	delimited	Decode
✓	⊖	delimited	Decode

Figure 3-24 Ignoring items using the Data tab

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

BIRT Analytics Loader supports creating multiple load transactions per project, as shown in Figure 3-25.



	Source	Database	Name	Description	N Columns
+	rdbms	SalesData	Customers		13
+	delimited	SalesData	CarInventory	Car Inventory and MSRP	3
+	delimited	SalesData	DailyPurchases		11
+	delimited	SalesData	MonthlySales		3
+	rdbms	SalesData	SaleOrders		7
+	rdbms	SalesData	Employees		8
+	delimited	SalesData	Clients		14

Figure 3-25 Creating multiple load transactions

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 allows 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 know how to do the following:

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

The example in "How to map data from a CSV file" provides more information about the mapping process.

Defining a target database

To define the target database and table, and the way the data is loaded, you must define the following information, as shown in Figure 3-26:

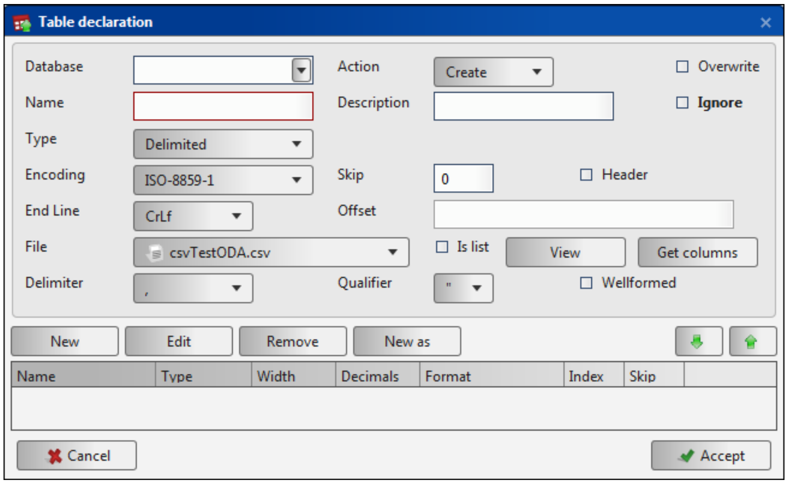


Table declaration

Database:
Name:
Type: Delimited
Encoding: ISO-8859-1
End Line: CrLf
File: csvTestODA.csv
Delimiter: ,
Action: Create
Description:
Skip: 0
Offset:
Qualifier: "
□ Overwrite
□ **Ignore**
□ Header
□ Is list
□ Wellformed

Name	Type	Width	Decimals	Format	Index	Skip

Figure 3-26 Creating a table mapping to a file

- Database
  Type a name for the target database, or select a listed database name.
- Name
  Type a unique name for a target data table.
- Description
  Optionally, type characters that describe the target data table.
- Action
  Select one of the following load operation types:
  - Create
    Adds a new table and overwrites any content stored in an existing table, using the displayed table name.
  - Append
    Adds records to any existing table that uses the displayed table name.
- Overwrite
  Choose this option when the loading transaction overwrites the target table.
- Ignore
  Choose this option when you must exclude the transaction from the load process.

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 line feed
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 Loader:

  ```
  ,  ->|  |  :  ;  @  #  "   +  -  =  '
  ```

- 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 Loader supports the following qualifier characters:

  ```
  "   '   ~
  ```

  Choose None, if the file format does not use a qualifier.

- Wellformed

  Select the Wellformed option to enforce verification that each record contains all columns.

  Wellformed explanation:

  - *"Checked"* - indicates the same number of columns in data source and table definition
  - *"Unchecked"* - extra columns are ignored and Null values are set for missing columns

Defining table columns

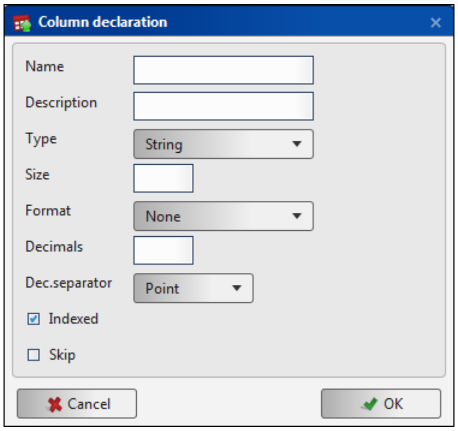
To automate defining the structure of a target data table, extract the data from a source data file, using Get columns. During extraction, Get columns adds column type and index information to each data element. You must review and edit, as necessary, the type and index information for every column. Consider removing the index from every column not strictly requiring an index. Removing unnecessary indexes speeds data load and analysis processes, but may limit performing aggregation and link operations. These operations require indexed columns.

- View

  Choose View to see the data in the file and check the connectivity to the file.

- Get columns
  Choose Get columns to get the column names and type automatically from the source file.

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, then use Edit to make modifications. To create a new column, choose New, then provide information using Column Declaration, as shown in Figure 3-27.


Column declaration

Name

Description

Type: String

Size

Format: None

Decimals

Dec.separator: Point

☑ Indexed

☐ Skip

Figure 3-27 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
    Enter the data type. 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 (-2,147,483,648) through 2^31 - 1 (2,147,483,647).
Longint	Integer (whole number) data from -2^63 (-9,223,372,036,854,775,808) through 2^63-1 (9,223,372,036,854,775,807).
Real	Floating precision number data with the following valid values: -3.40E + 38 through -1.18E - 38, 0 and 1.18E - 38 through 3.40E + 38.
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 yyyy_mm_dd_hh_MM_ss.
Time	The default format is hh_MM_ss.
Unicode	A sequence of Unicode characters, up to 1024.

- 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

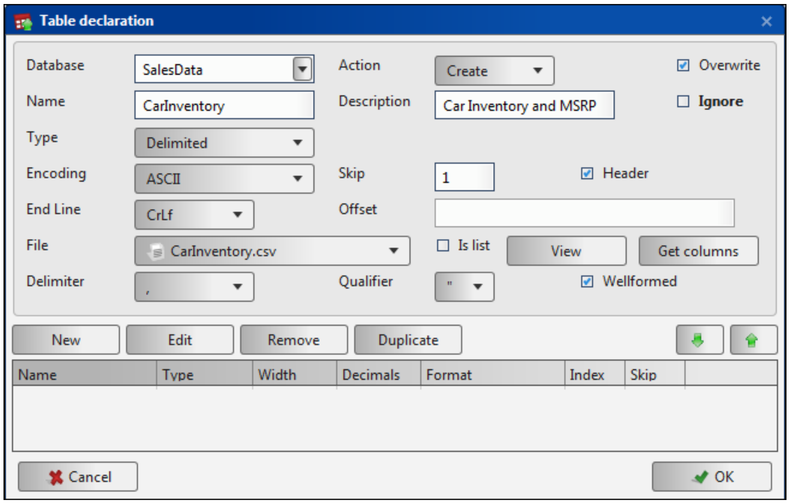
  Creates a new column by copying the properties of an existing column.

How to map data from a CSV file

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

```
PRODUCTNAME,QUANTITYINSTOCK,MSRP
1969 Harley Davidson Ultimate Chopper,7933,95.7
1952 Alpine Renault 1300,7305,214.3
```

1. In Options, choose New from file.
2. In Table Declaration, set properties for the target database, as shown in Figure 3-28.
   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 a target table to which you will map data. For example, CarInventory.
   3. In Description, type characters that describe information the table will contain, such as Car Inventory and MSRP.
   4. In Action, select Create.
   5. Select Overwrite.
3. Select options that match properties of the source file, as shown in Figure 3-28.
   1. In Type, select Delimited.
   2. In Encoding, select ASCII.
   3. Select Header. In this example, 1 appears in Skip. Edit the value in Skip to match the number of header rows in the source data file.
   4. In End Line, select CrLf.



Database	SalesData	Action	Create	☑ Overwrite	
Name	CarInventory	Description	Car Inventory and MSRP	□ Ignore	
Type	Delimited				
Encoding	ASCII	Skip	1	☑ Header	
End Line	CrLf	Offset			
File	CarInventory.csv	□ Is list	View	Get columns	
Delimiter	,	Qualifier	"	☑ Wellformed	

Name	Type	Width	Decimals	Format	Index	Skip

Figure 3-28 Defining file structure

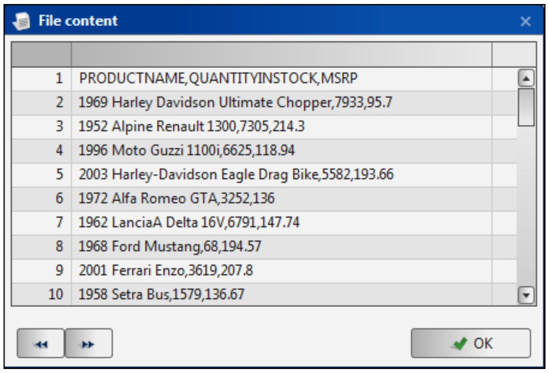
5 In File, select a source file name from the list.

6 In Delimiter, select comma (,).

7 In Qualifier, select ".

8 Select Wellformed.

4 Choose View to validate the data connection. The file content appears, as shown in Figure 3-29. Choose OK.



```
 1  PRODUCTNAME,QUANTITYINSTOCK,MSRP
 2  1969 Harley Davidson Ultimate Chopper,7933,95.7
 3  1952 Alpine Renault 1300,7305,214.3
 4  1996 Moto Guzzi 1100i,6625,118.94
 5  2003 Harley-Davidson Eagle Drag Bike,5582,193.66
 6  1972 Alfa Romeo GTA,3252,136
 7  1962 LanciaA Delta 16V,6791,147.74
 8  1968 Ford Mustang,68,194.57
 9  2001 Ferrari Enzo,3619,207.8
10  1958 Setra Bus,1579,136.67
```

Figure 3-29 Viewing file content

5 Choose Get columns to import the file metadata.

During the metadata import process, Get columns reads column name information from the header row in the source file, determines a type for each column, and indexes each column.

Definitions for each new column created in the target table appear listed in Table Declaration, as shown in Figure 3-30.

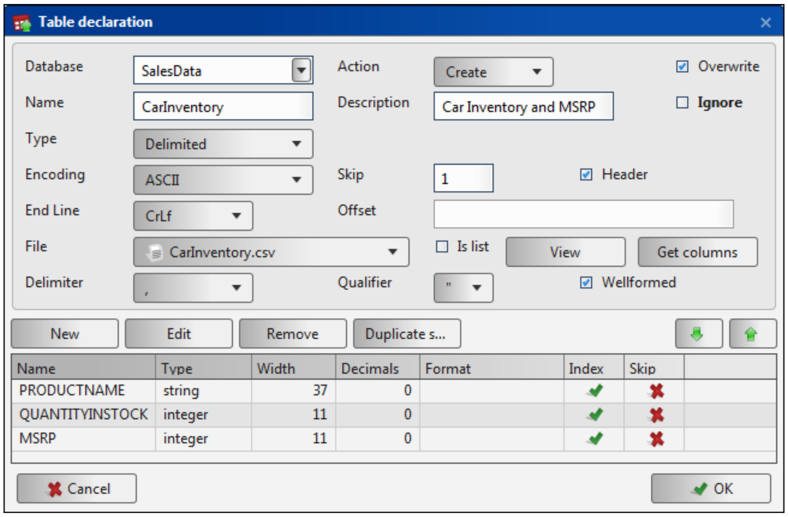
6 To modify properties for a column, select a listed column definition and choose Edit.

1 In Column Declaration, select or modify any available properties.

2 To remove index information from a column, deselect Indexed.

3 To exclude a column from loading, select Skip.

4 Choose OK.

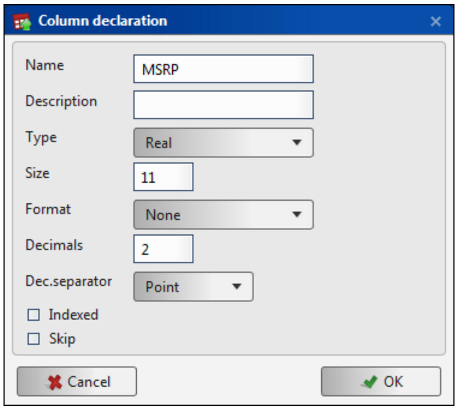


Database SalesData
Name CarInventory
Type Delimited
Encoding ASCII
End Line CrLf
File CarInventory.csv
Delimiter ,
Action Create
Description Car Inventory and MSRP
Skip 1
Offset
Qualifier "
☑ Overwrite
□ Ignore
☑ Header
□ Is list
☑ Wellformed

Name	Type	Width	Decimals	Format	Index	Skip
PRODUCTNAME	string	37	0		✓	✗
QUANTITYINSTOCK	integer	11	0		✓	✗
MSRP	integer	11	0		✓	✗

Figure 3-30 Getting column definitions

For example, change the type of the MSRP column from Integer to Real, and set the column to not indexed, as shown in Figure 3-31.

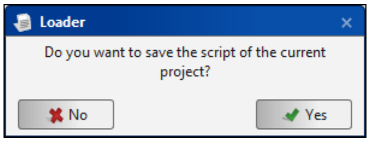


Name MSRP
Description
Type Real
Size 11
Format None
Decimals 2
Dec.separator Point
□ Indexed
□ Skip

Figure 3-31 Editing column definition properties

7 To close a selected database table, in Table Declaration, choose OK.

8 In Project, choose Save. Choose Yes to confirm saving the script for the current project, as shown in Figure 3-32.



Do you want to save the script of the current project?

Figure 3-32 Saving a 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 define the target database, provide a target table name, and choose a database data source. If necessary, to review the procedure for setting these properties, see “Defining a target database,” earlier in this chapter. Next, define the data set to retrieve from the source database using Table Declaration, as described in Defining a data set. Selecting a database in Datasource provides the option to define a query that retrieves a data set. “How to map data from an Oracle data source” provides more information about this process.

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 data source. You can type a query manually, and use Get columns to retrieve the columns from the data set.

  Alternatively, you can use View database structure to select tables and columns. Use Datasource to select tables and their columns, and specify the column order.
- Options
  Enter the options required by the connection drivers. For example, when loading data from an Oracle database, it is recommended to include 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 you edit the structure of a target data table, use Get columns to retrieve existing column information, then use Edit to make modifications. To create a new column, choose New, then provide information using Table Declaration, as shown in Figure 3-22.

How to map data from an Oracle data source

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

1 Define the target database and table.

1 In Options, choose New from DB. Table Declaration appears, as shown in Figure 3-33.

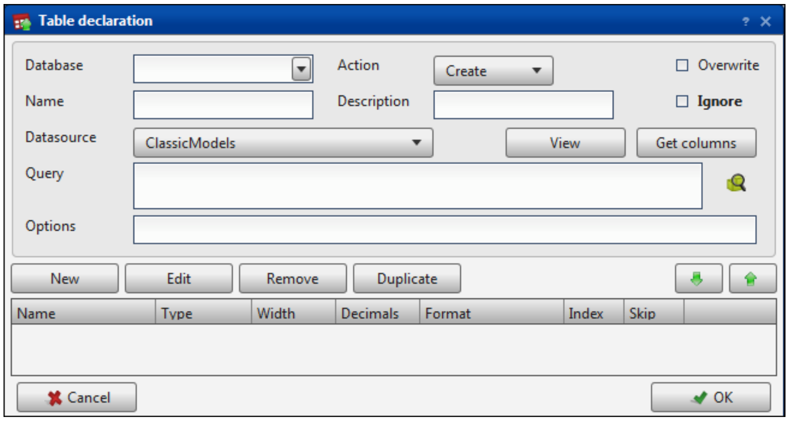


Table declaration

Database:
Action: Create
□ Overwrite
Name:
Description:
□ **Ignore**
Datasource: ClassicModels
Query:
Options:

Name	Type	Width	Decimals	Format	Index	Skip

Figure 3-33 Declaring a table from a database

2 In Table Declaration, choose the target database from the drop-down list, as shown in Figure 3-34.

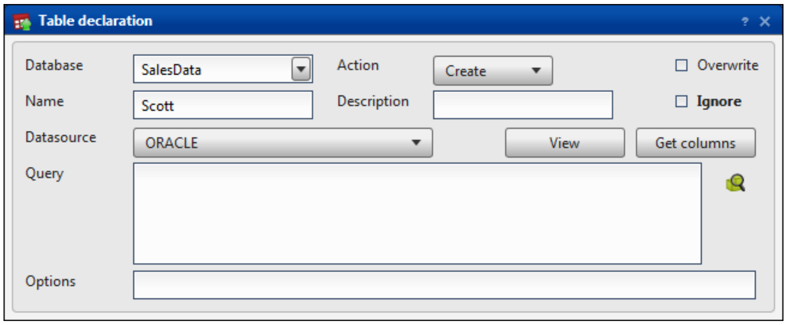


Table declaration

Database: SalesData
Action: Create
□ Overwrite
Name: Scott
Description:
□ **Ignore**
Datasource: ORACLE
Query:
Options:

Figure 3-34 Mapping data from an Oracle database

3 In Name, specify the name of the target table.

4 In Description, enter the table description.

5 In Action, choose Create.

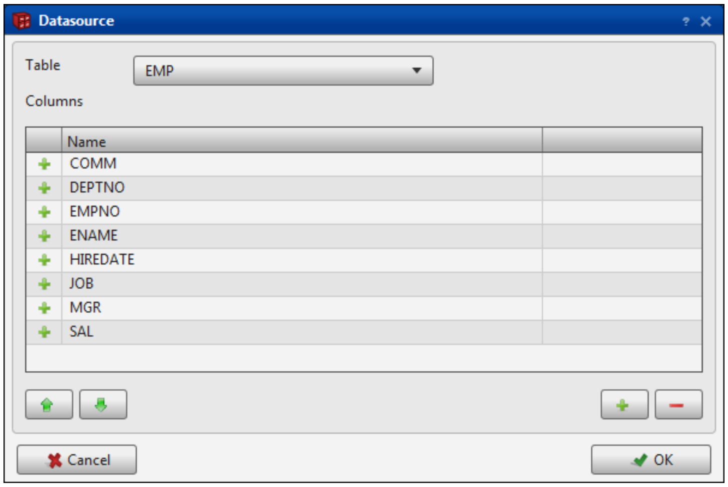
6 Select Overwrite.

2 Define the data set.

1 In Datasource, choose the database data source.

2 In Query, choose View database structure to build the query. Datasource appears.

- In Datasource, choose a table from the table drop-down list. The column names appear, as shown in Figure 3-35.
- 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. The column definitions appear, as shown in Figure 3-35.



Datasource

Table: EMP

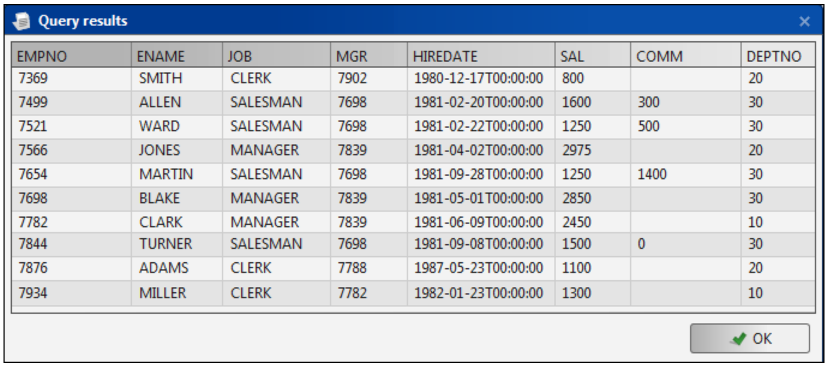
Columns

Name
COMM
DEPTNO
EMPNO
ENAME
HIREDATE
JOB
MGR
SAL

Figure 3-35 Using the query builder

3 In Options, specify any options you wish to pass to the data driver.

3 Choose View to validate the data connection and view the data. Query results appears, as shown in Figure 3-36.

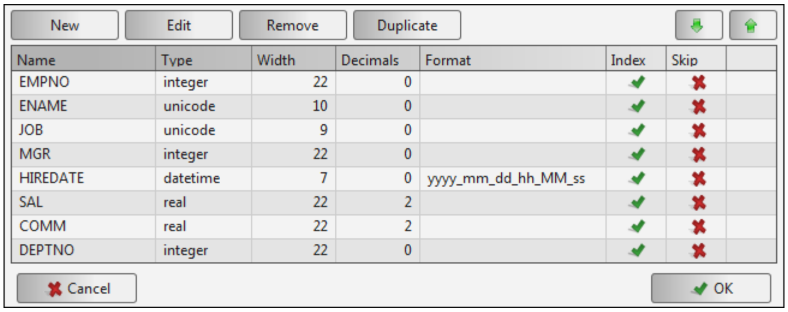


Query results

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	1980-12-17T00:00:00	800		20
7499	ALLEN	SALESMAN	7698	1981-02-20T00:00:00	1600	300	30
7521	WARD	SALESMAN	7698	1981-02-22T00:00:00	1250	500	30
7566	JONES	MANAGER	7839	1981-04-02T00:00:00	2975		20
7654	MARTIN	SALESMAN	7698	1981-09-28T00:00:00	1250	1400	30
7698	BLAKE	MANAGER	7839	1981-05-01T00:00:00	2850		30
7782	CLARK	MANAGER	7839	1981-06-09T00:00:00	2450		10
7844	TURNER	SALESMAN	7698	1981-09-08T00:00:00	1500	0	30
7876	ADAMS	CLERK	7788	1987-05-23T00:00:00	1100		20
7934	MILLER	CLERK	7782	1982-01-23T00:00:00	1300		10

Figure 3-36 Viewing data from a database

4 Choose Get columns to import the columns from the data set. The column definitions appear as shown in Figure 3-37. The types are determined automatically, and all columns are indexed by default. Get columns overwrites the existing data set, and discards your existing changes.

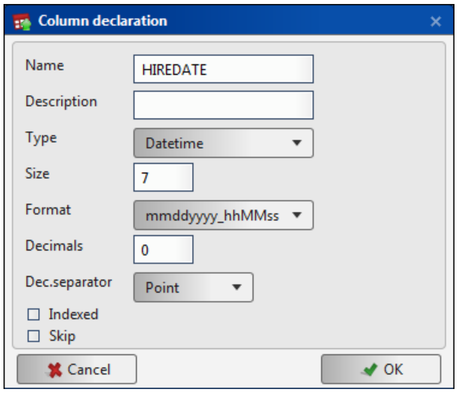


Name	Type	Width	Decimals	Format	Index	Skip
EMPNO	integer	22	0		✓	✗
ENAME	unicode	10	0		✓	✗
JOB	unicode	9	0		✓	✗
MGR	integer	22	0		✓	✗
HIREDATE	datetime	7	0	yyyy_mm_dd_hh_MM_ss	✓	✗
SAL	real	22	2		✓	✗
COMM	real	22	2		✓	✗
DEPTNO	integer	22	0		✓	✗

Figure 3-37 Getting the column definitions from the data set

5 Edit a column.

The example in Figure 3-38 shows changing the format of the HIREDATE column, and setting the column as not indexed.



Name: HIREDATE
Description:
Type: Datetime
Size: 7
Format: mmddyyyy_hhMMss
Decimals: 0
Dec.separator: Point
□ Indexed
□ Skip

Figure 3-38 Editing a Datetime column

1. Select the HIREDATE column and choose Edit.
2. In Type, select Date format.
3. In Format, select mm_dd_yyyy from the drop-down list.
4. Deselect Indexed. Choose OK.

6 In Project, choose Save. Choose Yes to save the project.

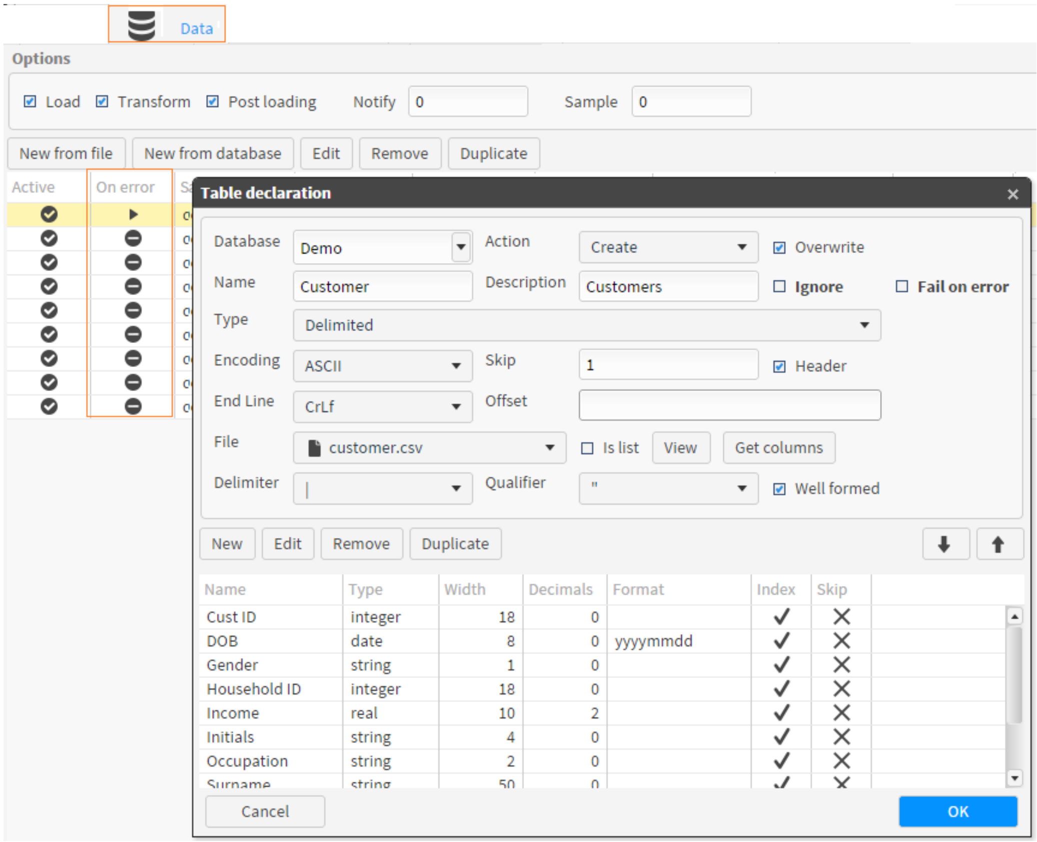
How to manage the loading process when exceptions occur

Users can decide *not to stop* the loading process when errors occur in the DATA tab. By default, the loading process stops whenever an error is encountered. However, thanks to a new *ONERROR attribute* in data table definition, it is now possible to choose whether you want the loading process to stop in the case of error or to continue loading. This allows the loading process to ignore any tables that present errors during the process, leaving them to be fixed and loaded later if desired. This can be particularly useful in the case of loading high volume tables needing overnight processing.

ON ERROR Attribute

In the DATA tab, clicking on 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.

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, checking the box once again. (See Figure 3-39).



☑ Load ☑ Transform ☑ Post loading Notify 0 Sample 0

Database: Demo · Action: Create · ☑ Overwrite

Name: Customer · Description: Customers · ☐ Ignore · ☐ Fail on error

Type: Delimited

Encoding: ASCII · Skip: 1 · ☑ Header

End Line: CrLf · Offset:

File: customer.csv · ☐ Is list

Delimiter: | · Qualifier: " · ☑ Well formed

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		✓	✕
Income	real	10	2		✓	✕
Initials	string	4	0		✓	✕
Occupation	string	2	0		✓	✕
Surname	string	50	0		✓	✕

Figure 3-39 Managing exceptions using the Data tab

Chapter

4

Managing data

This chapter contains:

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

About managing data

BIRT Loader Data Management, shown in Figure 4-1, provides tools for exploring data and creating and maintaining links between tables. BIRT Analytics Loader connects directly to the database engine. You explore the structure of all databases currently stored in FastDB, without synchronizing, using BIRT Analytics Loader Explorer. Explorer displays data objects, such as databases, tables, columns and their properties, and the discrete values of each column. The Links tool creates and maintains links between tables.

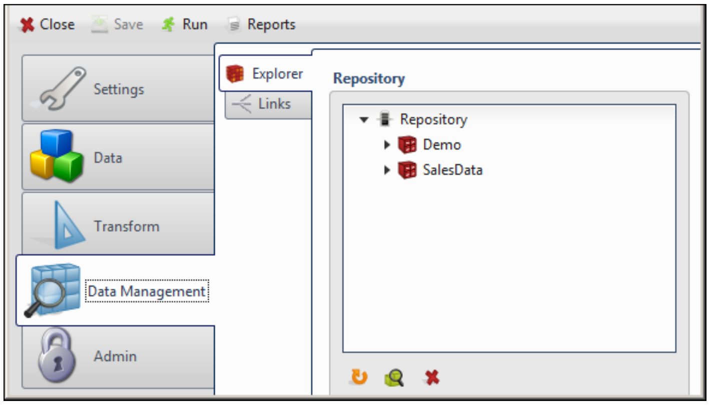


Figure 4-1 Managing data in FastDB

Understanding Explorer

Use Explorer to select and browse database structures stored in FastDB. Explorer supports browsing of multiple database objects, such as databases, tables, and columns. Explorer also supports refreshing the list of database names, viewing properties of a selected table and viewing properties of a selected column, using the features shown in Figure 4-2.

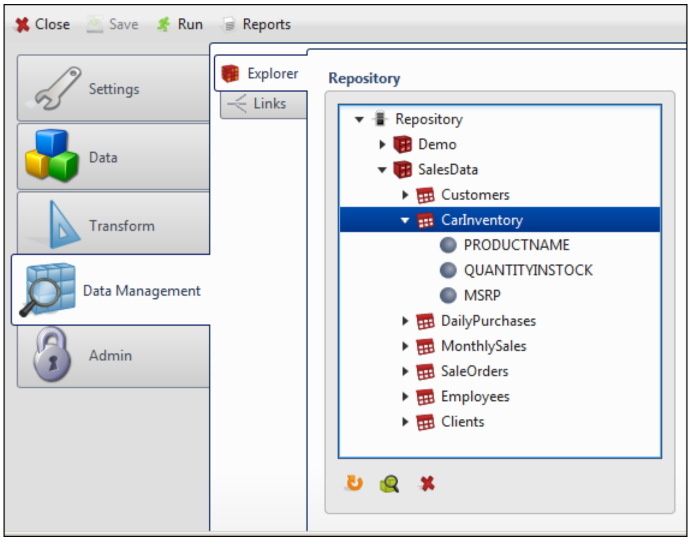


Figure 4-2 Exploring data structures in FastDB

How to view data structures stored in FastDB

1 In BIRT Analytics Loader, choose Data Management, and then Explorer.

The repository tree lists all database names currently stored in FastDB.

2 To view a list of tables in a database, in Repository, double-click a database name.

The database link expands and lists all table names defined in the selected database.

3 To view a list of database columns, in Repository, double-click a table name.

The table link expands and lists all column names defined in the selected table.

Refreshing the repository content

In Repository, choose Refresh to ensure that you see all databases, tables, and their columns currently stored in FastDB.

Exploring table properties

To view table properties, in Repository, select a table name, then choose View details. For a selected table, Table displays a value for each of the following table properties:

- ID
  Internal table identification number

- Name
  Table name

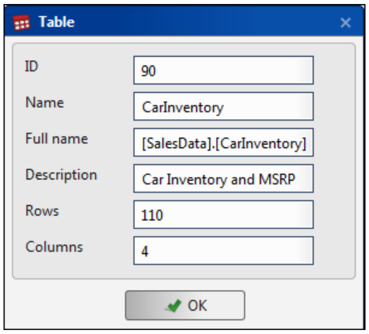
- Full name
  Full table name in format [Database].[Table]

- 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 CarInventory table in the SalesData database.



Table

ID: 90

Name: CarInventory

Full name: [SalesData].[CarInventory]

Description: Car Inventory and MSRP

Rows: 110

Columns: 4

Figure 4-3 Exploring table properties

Exploring column properties

To view column properties, in Repository, select a column name, then choose View details. For a selected column, Column displays Attributes and Discretes.

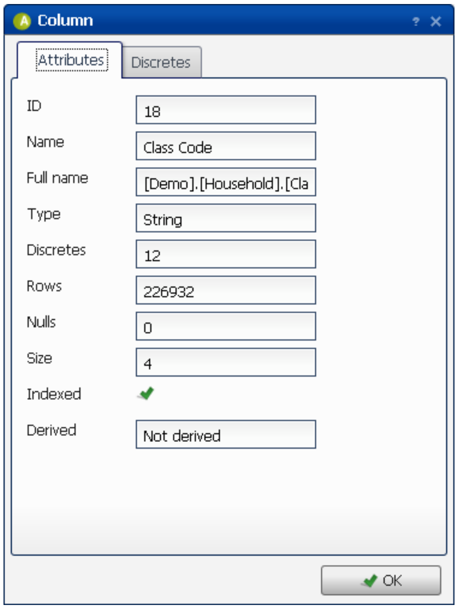
Attributes displays a value for each of the following column properties:

- 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 Class Code column in the Household table.



ID	18
Name	Class Code
Full name	[Demo].[Household].[Cla
Type	String
Discretes	12
Rows	226932
Nulls	0
Size	4
Indexed	✓
Derived	Not derived

Figure 4-4 Exploring properties of a permanent column

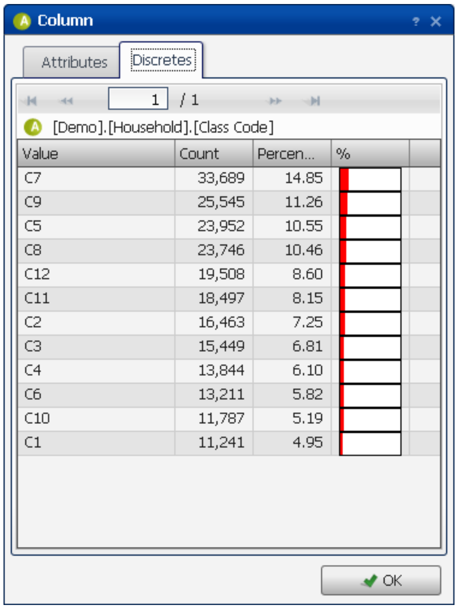
Discretes displays a value for each of the following column properties:

- 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 Class Code column in the Household table.



Value	Count	Percen...	%
C7	33,689	14.85	
C9	25,545	11.26	
C5	23,952	10.55	
C8	23,746	10.46	
C12	19,508	8.60	
C11	18,497	8.15	
C2	16,463	7.25	
C3	15,449	6.81	
C4	13,844	6.10	
C6	13,211	5.82	
C10	11,787	5.19	
C1	11,241	4.95	

Figure 4-5 Exploring discrete values of a permanent column

Dropping database objects

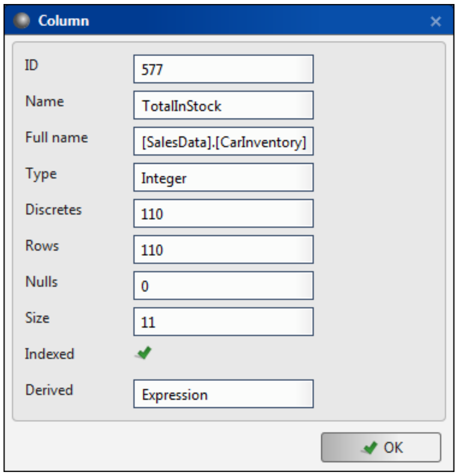
To drop databases, tables, or data columns, select the data object in the repository tree, and then choose Drop. Drop removes the data object from the FastDB database.

Understanding dynamic and permanent columns

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 using instructions known as transformation instructions. The dynamic columns appear as editable columns in BIRT Analytics.

For a dynamic column, the column property Derived displays the formula name of the translation instruction that you use to create the column. For example, Figure 4-6 shows the column properties for a dynamic column, [TotalInStock]. It was created using the translation instruction, Expression. The Derived property displays Expression.

BIRT Analytics uses the term permanent column to describe a column that contains data values. You typically load the data values in a permanent column from an external database. You can transform a dynamic column to a permanent column using the transformation instruction, MAKEPERMANENT. For more information about transformations, see Chapter 5, “Transforming data.”



Column

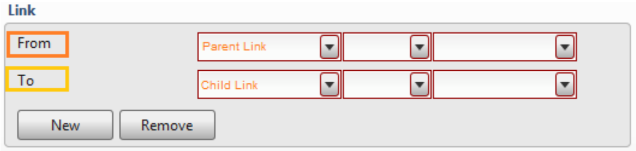
ID	577
Name	TotalInStock
Full name	[SalesData].[CarInventory]
Type	Integer
Discretes	110
Rows	110
Nulls	0
Size	11
Indexed	✓
Derived	Expression

Figure 4-6 Exploring properties for a derived column

Understanding Links

You link tables so that records from one table match related records from another. When you link, you are using a column that is common to both tables. BIRT Analytics uses the link to match up records from one table with those from the other. For example, if you use an Orders table and a Customers table, you link the tables so that each order from the Orders table can be matched up with the customer from the Customers table that made the order. In this example, the link assures that the data in each row of the report refers to the same order.

In the case of links that have a *1 to 1 relationship*, the link column indicated in the “From” field will automatically become the Parent table and the link column indicated in the “To” field will be the Child table. (See Figure 4-7 below).



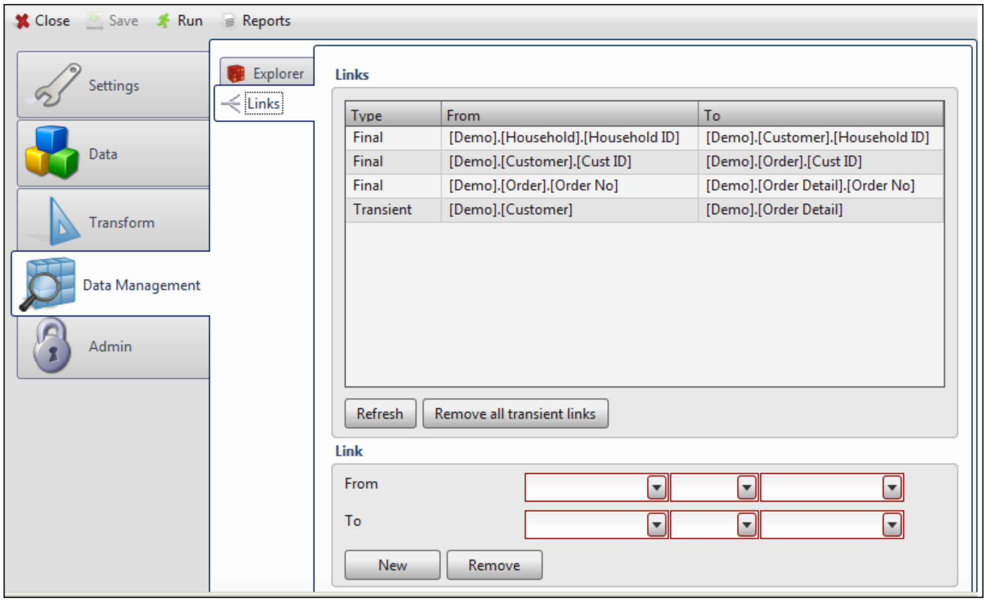
Link

From	Parent Link
To	Child Link

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

Links, shown in Figure 4-8, display final and transient links. Final 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 below, gets created if you create a Crosstab with [Customer].[Occupation] as a dimension and Count(Order Detail) as a measure.

To clear the transient links, choose Remove all transient links.



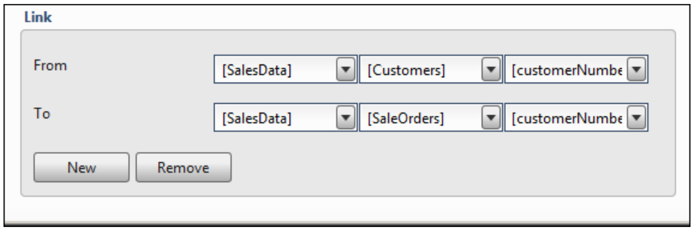
Type	From	To
Final	[Demo].[Household].[Household ID]	[Demo].[Customer].[Household ID]
Final	[Demo].[Customer].[Cust ID]	[Demo].[Order].[Cust ID]
Final	[Demo].[Order].[Order No]	[Demo].[Order Detail].[Order No]
Transient	[Demo].[Customer]	[Demo].[Order Detail]

Figure 4-8 Managing links

How to create a new link

1 In Link, in From, select the database, table, and column from which the link starts.

2 In Link, in To, choose the database, table, and column to which the link goes, as shown in Figure 4-9.



From: [SalesData] [Customers] [customerNumbe

To: [SalesData] [SaleOrders] [customerNumbe

Figure 4-9 Creating links

3 Choose New to create the link. Choose Yes in the message asking you to confirm. The new link appears in the list of links.

How to remove a link

1 In Link, in From, select the database, table, and column from which the link starts.

2 In Link, in To, choose the database, table, and column to which the link goes.

3 Choose Remove to drop the link. Choose Yes in the message asking you to confirm. The link disappears from the list of links.

Chapter

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 cleansing, aggregating, and integrating data from multiple sources. Transforming the data is executed after the load process is finished. The transformation script consists of one or more script files. The list of the script files is shown in Files, as shown in Figure 5-1. The Main file is the first file executed after data load. To access this file, you select Main file.

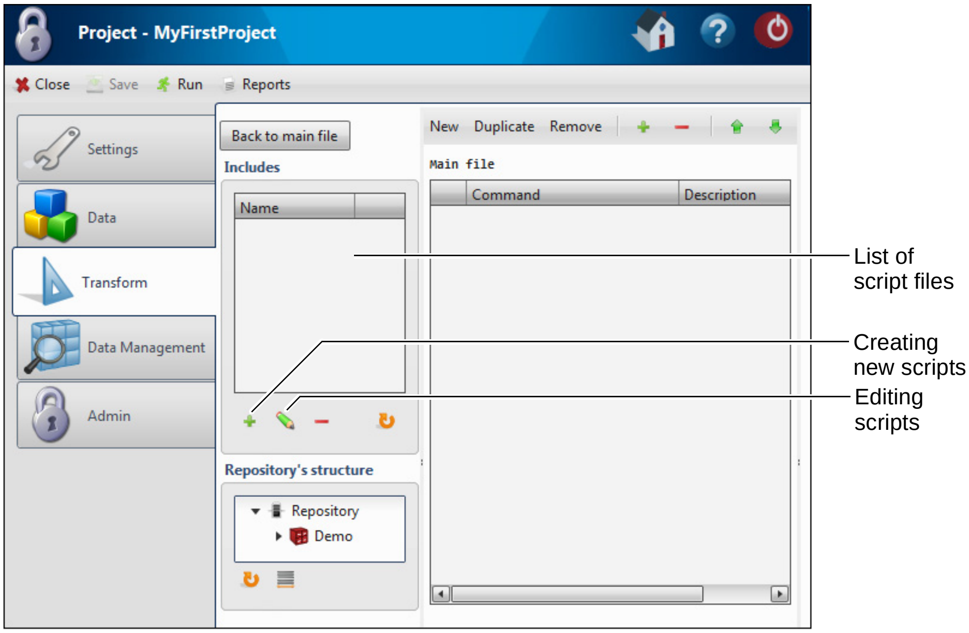
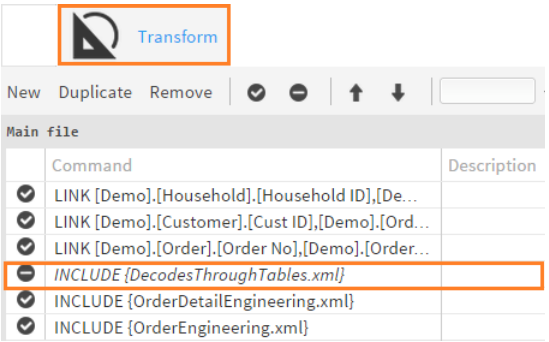


Figure 5-1 Transforming data

A new form for creating a new transaction opens on the right side of the Main file area. You can resize each of the three areas on the Transform screen by positioning your pointer on the three-dot line separating the areas, and dragging the pointer in the desired direction.

Ignoring items during the transform process

Users can choose to have items (tables, instructions) ignored during the transform process in the Transform tab (items can also be ignored during the loading process using the Data tab). Clicking on the "check" symbols in the column beside the "Command" column in the Transform tab allows Users to have chosen elements ignored by the transform process (toggling them on and off). The contents of ignored item rows appear in italics.



Main file

	Command	Description
✓	LINK [Demo].[Household].[Household ID],[De...	
✓	LINK [Demo].[Customer].[Cust ID],[Demo].[Ord...	
✓	LINK [Demo].[Order].[Order No],[Demo].[Order...	
⊖	*INCLUDE {DecodesThroughTables.xml}*	
✓	INCLUDE {OrderDetailEngineering.xml}	
✓	INCLUDE {OrderEngineering.xml}	

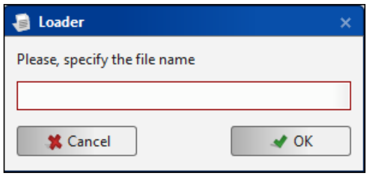
Figure 5-2 ignoring items using the Transform tab

Understanding Includes

You use Includes 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 5-3.



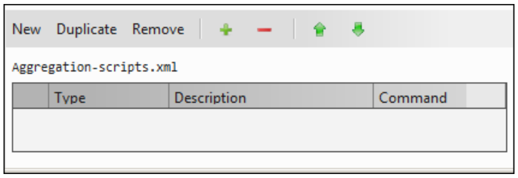
Loader

Please, specify the file name

Figure 5-3 Creating a new script

✎

- Edits a script file
Selecting Edit opens the file in the editing script area as shown in Figure 5-4.



Aggregation-scripts.xml

Type	Description	Command

Figure 5-4 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

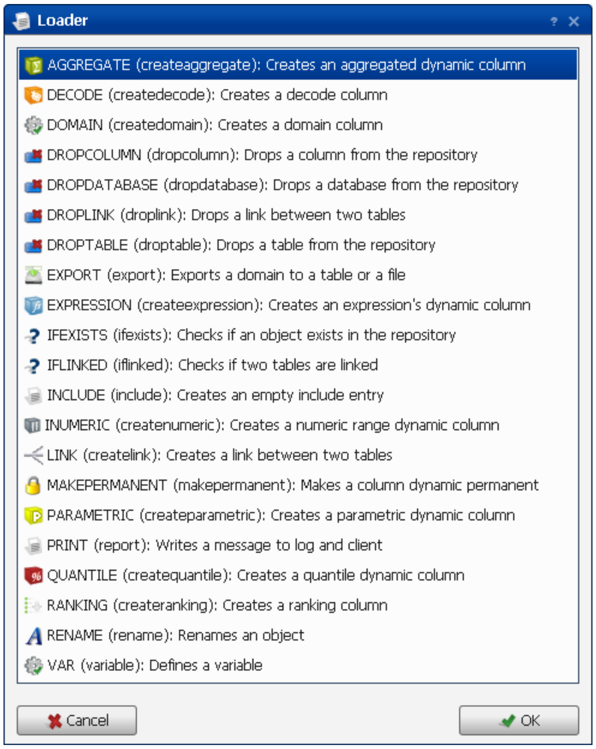
↻

- Refreshes the file list in Includes

Understanding scripts

Choosing Main file opens the main script in the editing area. Each script file contains transformation instructions. To create these instructions you use the following options:

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



AGGREGATE (createaggregate): Creates an aggregated dynamic column
DECODE (createdecode): Creates a decode column
DOMAIN (createdomain): Creates a domain column
DROPCOLUMN (dropcolumn): Drops a column from the repository
DROPDATABASE (dropdatabase): Drops a database from the repository
DROPLINK (droplink): Drops a link between two tables
DROPTABLE (droptable): Drops a table from the repository
EXPORT (export): Exports a domain to a table or a file
EXPRESSION (createexpression): Creates an expression's dynamic column
IFEXISTS (ifexists): Checks if an object exists in the repository
IFLINKED (iflinked): Checks if two tables are linked
INCLUDE (include): Creates an empty include entry
INUMERIC (createnumeric): Creates a numeric range dynamic column
LINK (createlink): Creates a link between two tables
MAKEPERMANENT (makepermanent): Makes a column dynamic permanent
PARAMETRIC (createparametric): Creates a parametric dynamic column
PRINT (report): Writes a message to log and client
QUANTILE (createquantile): Creates a quantile dynamic column
RANKING (createranking): Creates a ranking column
RENAME (rename): Renames an object
VAR (variable): Defines a variable

Figure 5-5 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 the transformation instructions, see *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
  Enables a search for an instruction

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

Understanding Repository

The repository structure, shown in Figure 5-6, displays the databases, their tables, and columns. Use Repository to drag and drop column definitions when you create filters in transformation instructions.

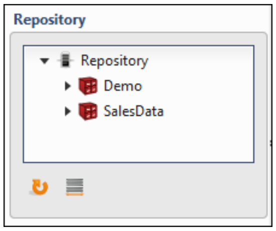
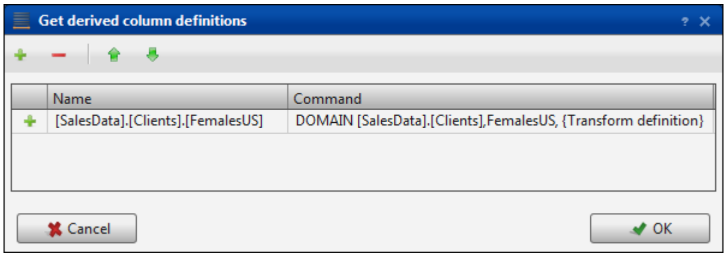


Figure 5-6 Repository structure

- Refresh the repository content.
- Get derived column definitions.
  Displays a list of the derived columns in the selected database, as shown in Figure 5-7.



	Name	Command
+	[SalesData].[Clients].[FemalesUS]	DOMAIN [SalesData].[Clients],FemalesUS, {Transform definition}

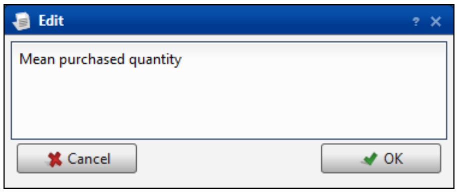
Figure 5-7 Getting a derived column definition

- Double-click a table column to explore the column properties and discrete values.

Typically, analysts create derived (calculated) columns during the data analysis. BIRT Loader administrators have to add these columns to the tables when refreshing the data during the load process. Otherwise, the definitions of these columns are lost, and must be recreated. To add the definitions to a script, copy instructions from the list and paste them to the script in Main file.

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.



Mean purchased quantity

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. 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 allows you to group information from different tables. For example, you can group information about orders in a table that contains customer data in order 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.

An Aggregate instruction must have the following syntax:

Syntax

```
AGGREGATE [Target table][Target column][Source column][Function]([Filter])
```

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:

```
/ \ ° ª - accents, dieresis,?, !, *, @, #, &, $, o, ñ
```

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

Filter

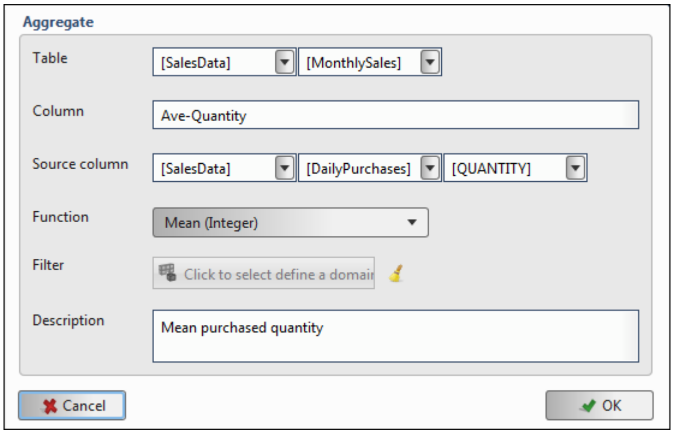
Use Domain definition to define a filter.

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 Main File, 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.



Aggregate

Table: [SalesData] [MonthlySales]

Column: Ave-Quantity

Source column: [SalesData] [DailyPurchases] [QUANTITY]

Function: Mean (Integer)

Filter: Click to select define a domain

Description: Mean purchased quantity

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.

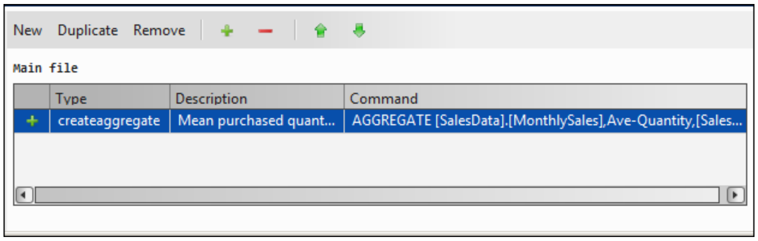
5 In Source column, select the database, the table, and the column you want to use to create the new column.

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 Main file, as shown in Figure 5-10.



Main file

	Type	Description	Command
+	createaggregate	Mean purchased quant...	AGGREGATE [SalesData].[MonthlySales],Ave-Quantity,[Sales...

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 that contains up to 100 discrete values. 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.

A Decode instruction must have the following syntax:

Syntax

```
DECODE [Source column][Target column][File][Code position]
   [Decode position][Encoding] [End line][Delimiter][Qualifier]
```

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:

```
/ \ ° ª - accents, dieresis,?, !, *, @, #, &, $, o, ñ
```

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 File Upload in Data tab to upload the decoding file and make it available in BIRT Analytics Loader.

Code position
The position of a code in the file.

Decode position
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.

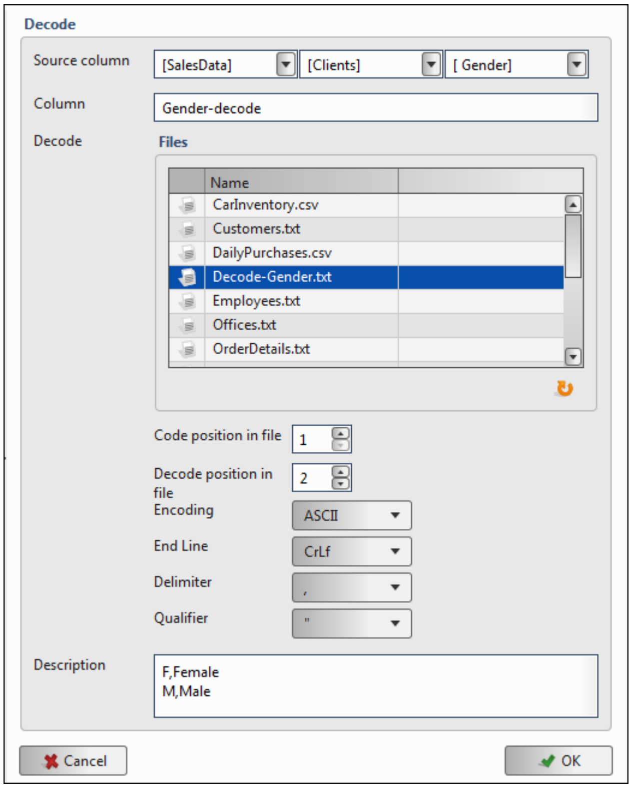
Qualifier
Surrounds the values in the file. Select a qualifier from the list of supported characters.

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 Main File, 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.



Decode

Source column: [SalesData] [Clients] [Gender]

Column: Gender-decode

Decode

Files

Name
CarInventory.csv
Customers.txt
DailyPurchases.csv
Decode-Gender.txt
Employees.txt
Offices.txt
OrderDetails.txt

Code position in file: 1

Decode position in file: 2

Encoding: ASCII

End Line: CrLf

Delimiter: ,

Qualifier: "

Description:
F,Female
M,Male

Figure 5-11 Creating a decoding column

3 In Source column, select a database, table, and column name from the drop-down lists.

4 In Column, type a name for the new column.

5 In Files, 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:

```
M,Male
F,Female
```

6 In Code position in file, type or select the number of the code column in the file. In this example, the position of the code is 1.

7 In Decode position in file, type or select the position of the decoding column. It is 2 in this example.

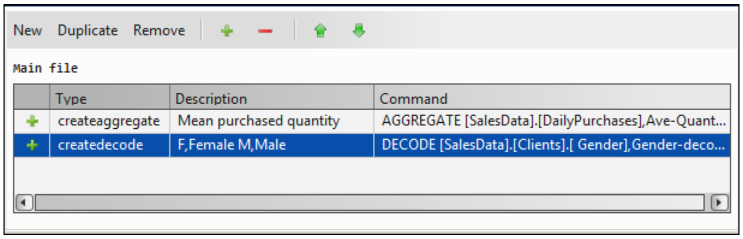
8 In End line, select the end line character used in the file from the drop-down list.

9 In Delimiter, select the delimiter used in the file from the drop-down-list.

10 In Qualifier, select the qualifier from the list.

11 In Description, enter a description of the column.

12 In Decode, select OK. The Decode instruction appears in Main file, as shown in Figure 5-12.



Main file

	Type	Description	Command
+	createaggregate	Mean purchased quantity	AGGREGATE [SalesData].[DailyPurchases],Ave-Quant...
+	createdecode	F,Female M,Male	DECODE [SalesData].[Clients].[Gender],Gender-deco...

Figure 5-12 Creating a Decode instruction

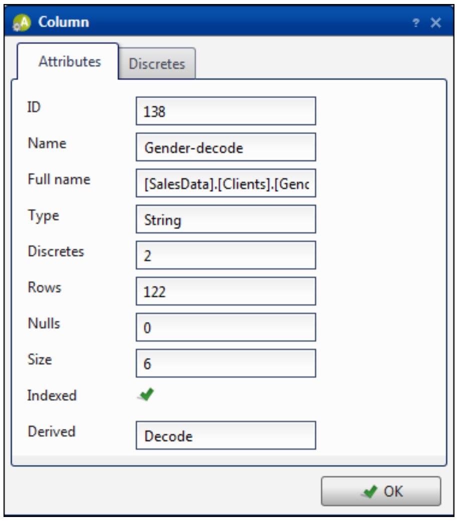
13 Choose Save to save the project.

14 Validate the new column was created.

1 Choose Data Management, and then Explorer.

2 In Database, select your database, and expand the list of tables.

3 In Tables, 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.



ID	138
Name	Gender-decode
Full name	[SalesData].[Clients].[Genc
Type	String
Discretes	2
Rows	122
Nulls	0
Size	6
Indexed	✓
Derived	Decode

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.

Domain can be defined as a simple query, or as a more complex set of segments. A Domain instruction must have the following syntax:

Syntax

```
DOMAIN [Target table] [Target column] [Domain]
```

Parameters

Target table
Defines a target table.

Target column
Defines a target column. Do not use the following characters:

```
/ \ ° ª - accents, dieresis, ?, !, *, @, #, &, $, o, ñ
```

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.
	New block	Defines a new block of segments. A segment may contain multiple queries. A domain can be build of multiple blocks. The blocks act as parenthesis and define the order of execution.
	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 check 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.

You can find more information about domain features in *Using BIRT Analytics*.

How to create a domain

1. In Main File, choose New, and then Domain in the list of transformation instructions. Domain appears.
2. In Table, select the database, and the table from the drop-down list, or select the table in Repository and drag it to the column box.

3 In Column, type the new column name, as shown in Figure 5-14.

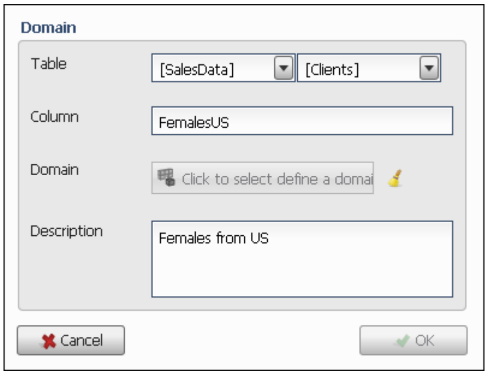


Table: [SalesData] [Clients]

Column: FemalesUS

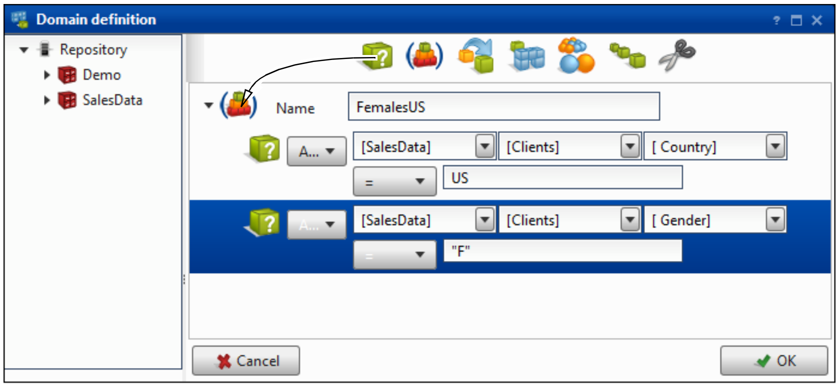
Domain: Click to select define a domai

Description: Females from US

Figure 5-14 Creating a domain

4 In Domain, choose Click to select define a domain. Domain definition appears.

5 Create a new query, as shown in Figure 5-15.



Name: FemalesUS

A... [SalesData] [Clients] [Country] = US

A... [SalesData] [Clients] [Gender] = "F"

Figure 5-15 Defining a domain

1 In Name, type the name of the domain.

2 Drag New query from the toolbar over New block, located on the left side of Name, as shown in Figure 5-15.

3 In New query, select a database, a table, and a column, or drag a column from Repository.

4 In New query, select the operator, for example =, from the drop-down list.

5 Type a value, for example US.

6 Drag a New query to the New block to add a second condition.

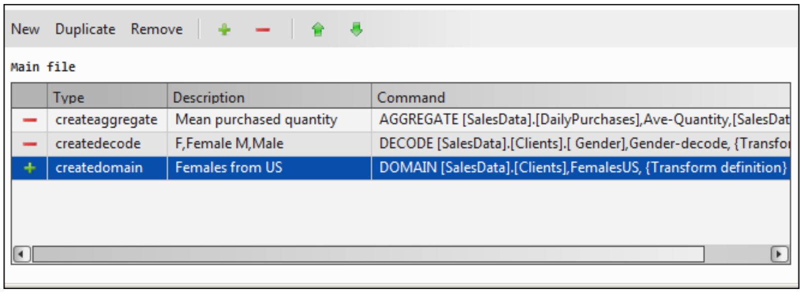
7 Select the operator between the two conditions. If you wish to get the records for all women in US, you must select AND.

8 Select a database, a table, and a column for the second condition.

9 Select operator, and type or select a value for the comparison.

10 In Domain definition, choose OK.

6 In Domain, choose OK. The new domain instruction appears in Main file, as shown in Figure 5-16.



Main file

	Type	Description	Command
−	createaggregate	Mean purchased quantity	AGGREGATE [SalesData].[DailyPurchases],Ave-Quantity,[SalesDat
−	createdecode	F,Female M,Male	DECODE [SalesData].[Clients].[Gender],Gender-decode, {Transfo
+	createdomain	Females from US	DOMAIN [SalesData].[Clients],FemalesUS, {Transform definition}

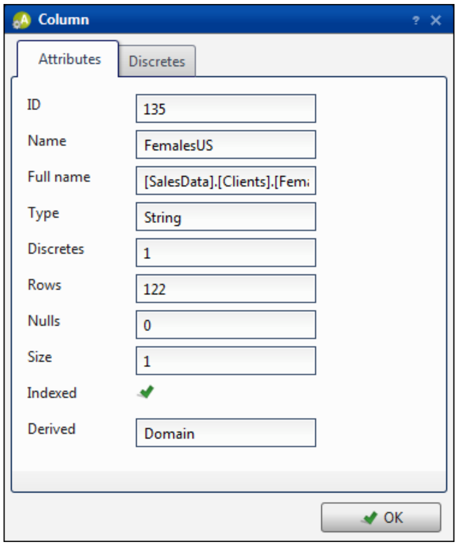
Figure 5-16 Creating a Domain instruction

7 Choose Save to save the project.

8 Validate the new column was created.

   1 Choose Data Management, and then Explorer.

   2 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.



ID	135
Name	FemalesUS
Full name	[SalesData].[Clients].[Fem
Type	String
Discretes	1
Rows	122
Nulls	0
Size	1
Indexed	✓
Derived	Domain

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. For expressions that generate a decimal value, you can specify precision up to six decimal places.

An Expression instruction must have the following syntax:

Syntax

```
EXPRESSION [Target table] [Target column] [Expression]
```

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:

```
/ \ ° ª - accents, dieresis, ?, !, *, @, #, &, $, o, ñ
```

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

Name and syntax	Description
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>)

(continues)

Table 5-5 Supported functions (continued)

Name and syntax	Description
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 number of 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 each surname stored in the column only once.
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.
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>)

Table 5-5 Supported functions (continued)

Name and syntax	Description
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 and / or time column.
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 (*n*). 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>)
REPLACE ([db].[table].[column], pattern, replacement)	Replaces the pattern text with the replacement text. Both pattern and replacement can be columns or text

(continues)

Table 5-5 Supported functions (continued)

Name and syntax	Description
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.
SECSTO ([db].[table].[column], [db].[table].[column])	Returns the seconds elapsed between two columns with date or time format.
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 *n*, where *n* 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.

Table 5-5 Supported functions (continued)

Name and syntax	Description
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.

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>)

VAL ([db].[table].[column])	Returns a Unicode string that represents any other type of data.
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.

Supported formats when working with DATE, TIME or DATETIME

The format is case sensitive, and these are the identifiers:

yyyy – represents the year with 4 figures

mm – represents the month with 2 figures

dd – represents the day with 2 figures

hh – represents the hour with 2 figures

MM – represents the minute with 2 figures

ss – represents the second with 2 figures

xxx – represents the millisecond with 3 figures

The character "_" represents any delimiter (any character)

The following formats are supported when working with the DATE, TIME or DATETIME instructions:

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

How to create an expression

This example creates a column that displays the total dollar amount of car inventory.

1 In Main file, 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.

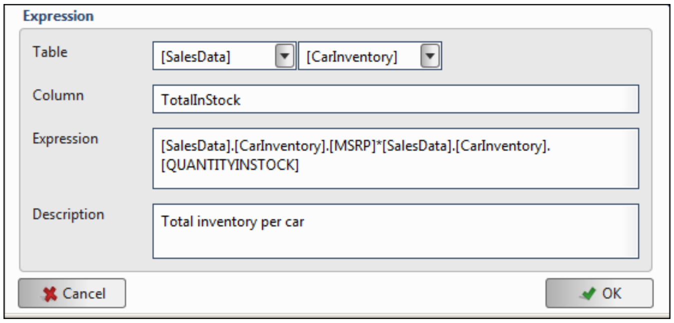


Table [SalesData] [CarInventory]

Column TotalInStock

Expression [SalesData].[CarInventory].[MSRP]*[SalesData].[CarInventory].[QUANTITYINSTOCK]

Description Total inventory per car

Figure 5-18 Creating an expression

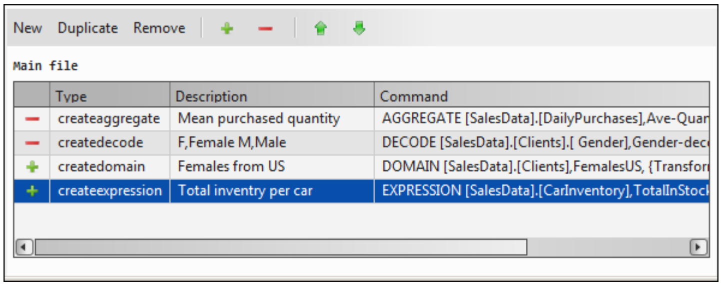
3 In Table, select a target database and a table.

4 In Column, type the name of the new column.

5 In Expression, type the expression.

6 In Description, type the description of the column.

7 In Expression, choose OK. The Expression instruction appears in Main file, as shown in Figure 5-19.



Main file

	Type	Description	Command
−	createaggregate	Mean purchased quantity	AGGREGATE [SalesData].[DailyPurchases],Ave-Quar
−	createdecode	F,Female M,Male	DECODE [SalesData].[Clients].[Gender],Gender-dec
+	createdomain	Females from US	DOMAIN [SalesData].[Clients],FemalesUS, {Transfor
+	createexpression	Total inventry per car	EXPRESSION [SalesData].[CarInventory],TotalInStoc

Figure 5-19 Creating an Expression instruction

8 Choose Save to save the project.

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

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. Depending on the type of analysis you want to do, it might be useful to 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.

An INumeric instruction must have the following syntax:

Syntax

```
INUMERIC [Source column] [Target column] [Bands] ([Filter])
```

Parameters

Source column
Defines a database, table, and column for a source for the numeric ranges. The column must be of numeric type.

Target column
Defines the name of the target column. The column is added to the source table.

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

```
1|30|Young
31|55|Middle-age
56|105|Old
```

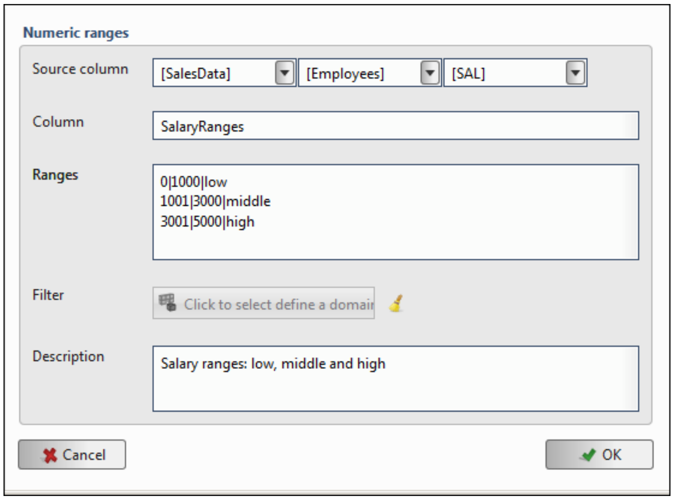
Filter

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

How to create numeric ranges

1 In Main file, choose New. The list of transformation instructions appears.

2 In the list of transformation instructions, select INUMERIC. Numeric ranges appears, as shown in Figure 5-20.



Numeric ranges

Source column: [SalesData] [Employees] [SAL]

Column: SalaryRanges

Ranges:

```
0|1000|low
1001|3000|middle
3001|5000|high
```

Filter: Click to select define a domair

Description: Salary ranges: low, middle and high

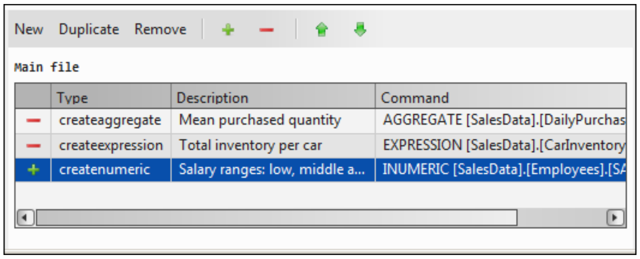
Figure 5-20 Creating numeric ranges

3 In Source column, 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 Column, type the name of the new numeric range column.

5 In Ranges, define the ranges and their names. Double-click in the box to open the text editor and type the text.

6 In Numeric ranges, choose OK. The INumeric instruction appears in Main file, as shown in Figure 5-21.



Main file

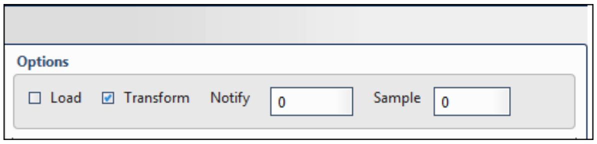
	Type	Description	Command
−	createaggregate	Mean purchased quantity	AGGREGATE [SalesData].[DailyPurchas
−	createexpression	Total inventory per car	EXPRESSION [SalesData].[CarInventory
+	createnumeric	Salary ranges: low, middle a...	INUMERIC [SalesData].[Employees].[SA

Figure 5-21 Creating an INumeric instruction

7 Choose Save to save the project.

8 Run the instruction.

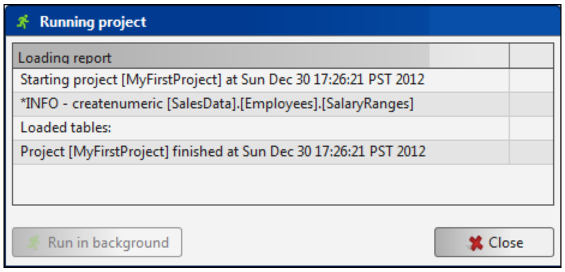
1. In Main file, select each instruction and then, -, to make it inactive, as shown in Figure 5-21.
2. In Data—Options, deselect Load, as shown in Figure 5-22.



☐ Load ☑ Transform Notify 0 Sample 0

Figure 5-22 Deselecting Load

3. Choose Run. As the project runs, messages describing each stage of the project appear in Running project, as shown in Figure 5-23.



Loading report
Starting project [MyFirstProject] at Sun Dec 30 17:26:21 PST 2012
*INFO - createnumeric [SalesData].[Employees].[SalaryRanges]
Loaded tables:
Project [MyFirstProject] finished at Sun Dec 30 17:26:21 PST 2012

Figure 5-23 Running the INumeric instruction

9 Validate 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 number of 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.

A Parametric instruction must have the following syntax:

Syntax

```
PARAMETRIC [Target table] [Target column] [Bins]
```

Parameters

Target table
Defines a database, and table.

Target column
Defines the name of the target column.

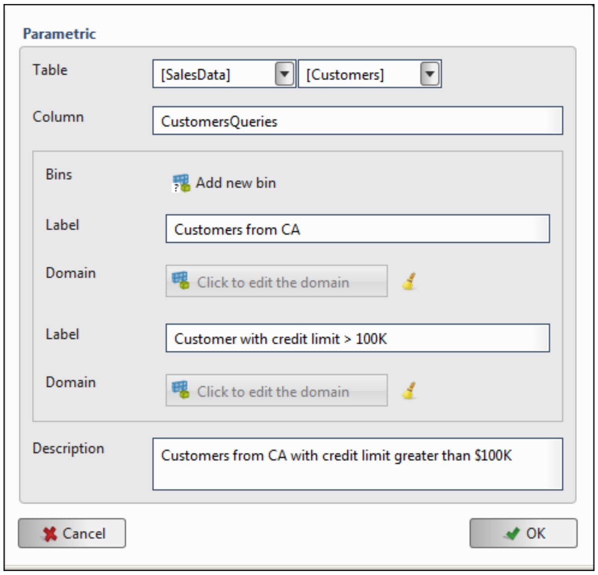
Bins
A set of data values that fall in a given interval. Define a bin name and a query that fills the bin with data. To define the query, use Domain.

How to create a parametric column

The example creates a parametric column, which groups the customers into two bins, customers from California, and customers with credit limit greater than $100K.

1 In Main file, choose New. The list of transformation instructions appears.

2 In the list of transformation instructions, choose Parametric. Parametric appears, as shown in Figure 5-24.



Parametric

Table: [SalesData] [Customers]

Column: CustomersQueries

Bins

Label: Customers from CA

Domain: Click to edit the domain

Label: Customer with credit limit > 100K

Domain: Click to edit the domain

Description: Customers from CA with credit limit greater than $100K

Figure 5-24 Creating a parametric column

3 In Table, select the target database and table.

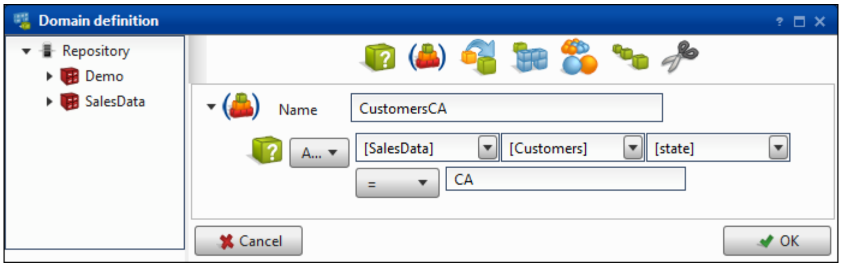
4 In Column, type the name of the new column.

5 Define a new bin.

1 In Bins, select Add new bin. The new bin definition appears.

2 In Label, type the name of the new bin, for example Customers from CA.

3 Choose Domain to define the query. Domain definition appears, as shown in Figure 5-25.



Domain definition

Name: CustomersCA

A... [SalesData] [Customers] [state]

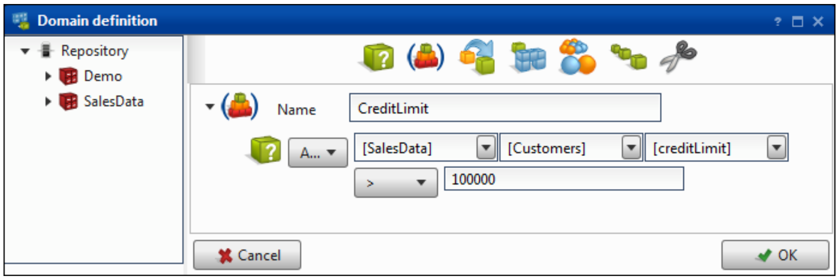
= CA

Figure 5-25 Creating the CustomersCA domain

4 In Domain definition, define the query and choose OK.

5 In Parametric, choose Add new bin again, to define another bin.

6 Define a label for the domain, and choose Domain to define the query. Domain definition appears again, as shown in Figure 5-26.



Name: CreditLimit

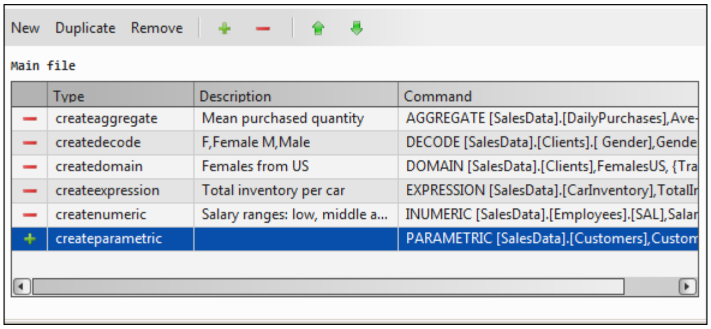
[SalesData] [Customers] [creditLimit]

> 100000

Figure 5-26 Creating the CreditLimit domain

7 In Domain definition, choose OK.

6 In Parametric, choose OK. The Parametric instruction appears, as shown in Figure 5-27.



Main file

	Type	Description	Command
—	createaggregate	Mean purchased quantity	AGGREGATE [SalesData].[DailyPurchases],Ave
—	createdecode	F,Female M,Male	DECODE [SalesData].[Clients].[Gender],Gende
—	createdomain	Females from US	DOMAIN [SalesData].[Clients],FemalesUS, {Tra
—	createexpression	Total inventory per car	EXPRESSION [SalesData].[CarInventory],TotalIr
—	createnumeric	Salary ranges: low, middle a...	INUMERIC [SalesData].[Employees].[SAL],Salar
+	createparametric		PARAMETRIC [SalesData].[Customers],Custom

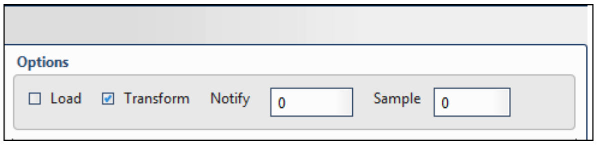
Figure 5-27 Creating a Parametric instruction

7 Choose Save, to save the project.

8 Run the instruction.

1 In Main file, select each instruction and then, choose - to make it inactive, as shown in Figure 5-21.

2 In Data—Options, deselect Load, as shown in Figure 5-28.

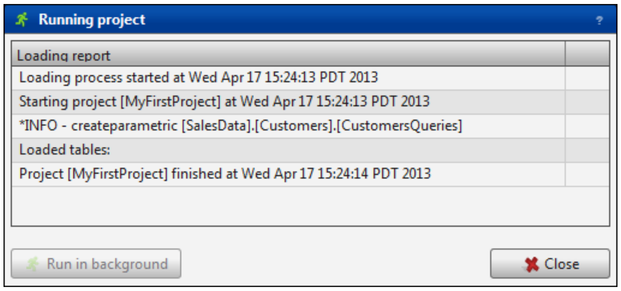


Options

□ Load ☑ Transform Notify 0 Sample 0

Figure 5-28 Deselecting Load

3 Choose Run. As the project runs, messages describing each stage of the project appear in Running project, as shown in Figure 5-29.



Loading report

Loading process started at Wed Apr 17 15:24:13 PDT 2013

Starting project [MyFirstProject] at Wed Apr 17 15:24:13 PDT 2013

*INFO - createparametric [SalesData].[Customers].[CustomersQueries]

Loaded tables:

Project [MyFirstProject] finished at Wed Apr 17 15:24:14 PDT 2013

Figure 5-29 Running a Parametric instruction

9 Validate 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.

A Quantile instruction must have the following syntax:

Syntax

```
QUANTILE [Source column] [Target column] [Number of parts] ([Filter])
```

Parameters

Source column
Defines a numeric column.

Target column
Defines the name of the new target column.

Number of parts
Defines the number of parts the data must be grouped into.

Filter
Defines a domain, if the data requires filtering.

How to create a quantile range column

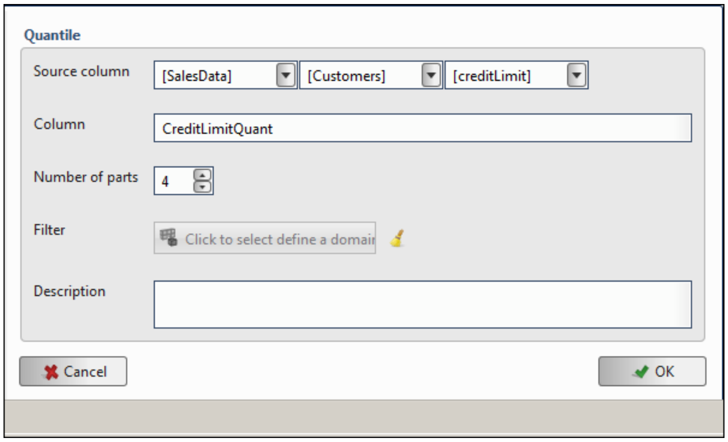
The following example describes how to create a quantile column that creates four groups that span the range of credit limit values found in a set of customer records. Table 5-6 lists the resulting quantile ranges and the number of customer records in each range.

Table 5-6 Quantile ranges

#	Credit limit range	Number of records
1	0-41900	31
2	43300-77000	31
3	77000-95100	30
4	95400-222600	30

1 In Main file, choose New. The list of transformation instructions appears.

2 In the list of transformation instructions, choose Quantile. Quantile appears, as shown in Figure 5-30.



Quantile

Source column: [SalesData] [Customers] [creditLimit]

Column: CreditLimitQuant

Number of parts: 4

Filter: Click to select define a domai

Description:

Figure 5-30 Creating a quantile column

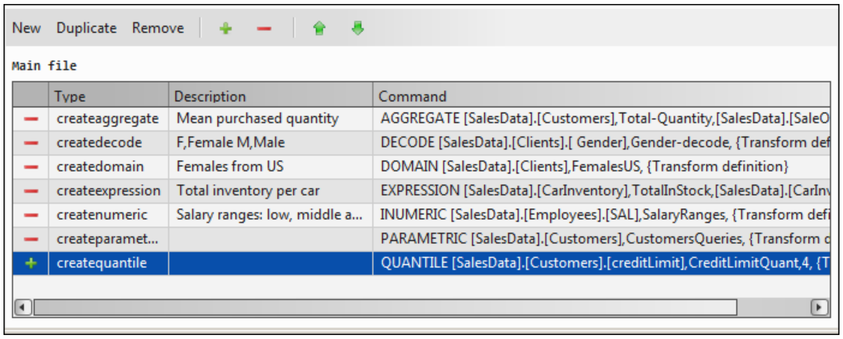
3 In Source column, select a database, a table, and a numeric column.

4 In Column, type the name of the new column.

5 In Number of parts, provide the number of quantiles, groups, or ranges that the column will contain.

6 In Filter, select Click to select define a domain, if you need to filter the data.

7 In Quantile, choose OK. The Quantile instruction appears in Main file, as shown in Figure 5-31.



Main file

	Type	Description	Command
−	createaggregate	Mean purchased quantity	AGGREGATE [SalesData].[Customers],Total-Quantity,[SalesData].[SaleO
−	createdecode	F,Female M,Male	DECODE [SalesData].[Clients].[Gender],Gender-decode, {Transform def
−	createdomain	Females from US	DOMAIN [SalesData].[Clients],FemalesUS, {Transform definition}
−	createexpression	Total inventory per car	EXPRESSION [SalesData].[CarInventory],TotalInStock,[SalesData].[CarInv
−	createnumeric	Salary ranges: low, middle a...	INUMERIC [SalesData].[Employees].[SAL],SalaryRanges, {Transform defi
−	createparamet...		PARAMETRIC [SalesData].[Customers],CustomersQueries, {Transform d
+	createquantile		QUANTILE [SalesData].[Customers].[creditLimit],CreditLimitQuant,4, {T

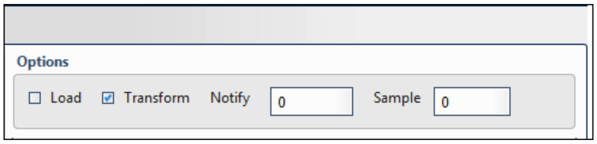
Figure 5-31 Creating a Quantile instruction

8 Save the project.

9 Run the instruction.

1 In Main file, select each instruction and then, choose - to make it inactive, as shown in Figure 5-21.

2 In Data—Options, deselect Load, as shown in Figure 5-32.

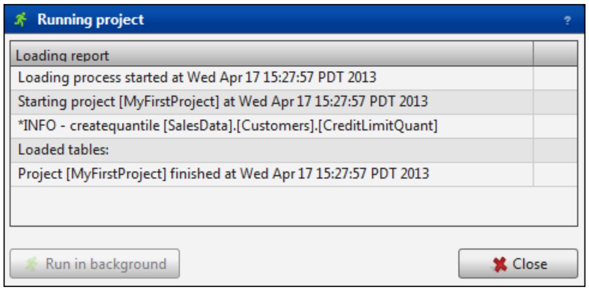


Options

☐ Load ☑ Transform Notify 0 Sample 0

Figure 5-32 Deselecting Load

3 Choose Run. As the project runs, messages describing each stage of the project appear in Running project, as shown in Figure 5-33.



Loading report
Loading process started at Wed Apr 17 15:27:57 PDT 2013
Starting project [MyFirstProject] at Wed Apr 17 15:27:57 PDT 2013
*INFO - createquantile [SalesData].[Customers].[CreditLimitQuant]
Loaded tables:
Project [MyFirstProject] finished at Wed Apr 17 15:27:57 PDT 2013

Figure 5-33 Running a Quantile instruction

RANKING

Supports creating a new column with the rank for each subset of records relating to the same key. Ranking allows 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. The instruction syntax is:

Syntax

```
RANKING [Key to rank] [Column to use] [Column name] [Sorting order] ([Force
   sequence] [Filter])
```

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.

Sorting direction
Defines the sorting direction, ASC or DESC.

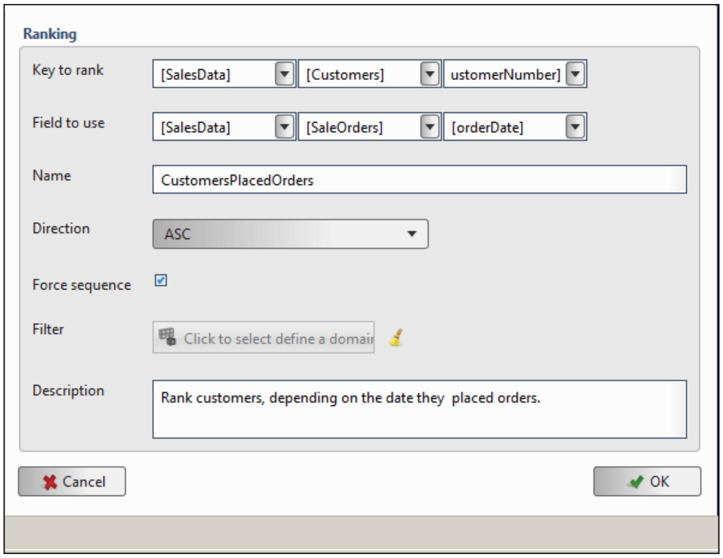
Force sequence
Defines the ranking sequence. In cases where two or more records have exactly the same value, checking Force sequence ensures that each record is allocated a different rank. If you do not check Force sequence, records with the same value have the same rank.

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

How to create a ranking column

1 In Main file, choose New. The list of transformation instructions appears.

2 In the list of transformation instructions, choose Ranking. Ranking appears, as shown in Figure 5-34.



Ranking

Key to rank: [SalesData] [Customers] ustomerNumber]

Field to use: [SalesData] [SaleOrders] [orderDate]

Name: CustomersPlacedOrders

Direction: ASC

Force sequence: ☑

Filter: Click to select define a domair

Description: Rank customers, depending on the date they placed orders.

Figure 5-34 Creating a ranking column

3 In Key to rank, select a database, a table, and a column you use as a key for ranking.

4 In Field to use, select a database, a table, and a column you use to determine the ranking values.

5 In Name, type the name of the new ranking column.

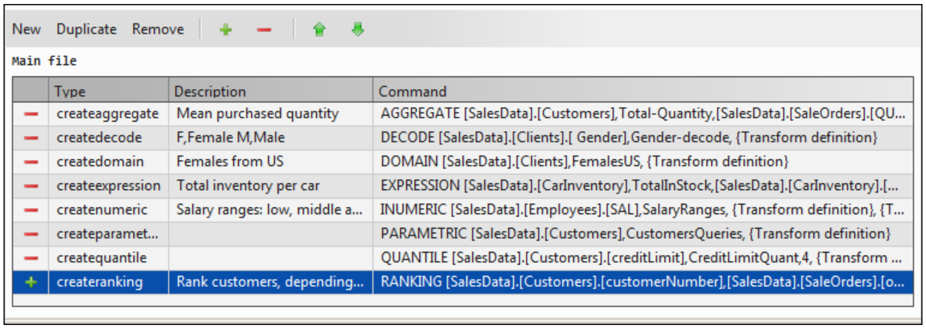
6 In Direction, select the table-sorting direction from the drop-down list.

7 Select Force sequence.

8 In Filter, choose Click to select define a domain, if you wish to filter the data.

9 In Description, type a short description of the column.

10 Choose OK. The Ranking instruction appears in Main file, as shown in Figure 5-35.



Main file

	Type	Description	Command
—	createaggregate	Mean purchased quantity	AGGREGATE [SalesData].[Customers],Total-Quantity,[SalesData].[SaleOrders].[QU...
—	createdecode	F,Female M,Male	DECODE [SalesData].[Clients].[Gender],Gender-decode, {Transform definition}
—	createdomain	Females from US	DOMAIN [SalesData].[Clients],FemalesUS, {Transform definition}
—	createexpression	Total inventory per car	EXPRESSION [SalesData].[CarInventory],TotalInStock,[SalesData].[CarInventory].[...
—	createnumeric	Salary ranges: low, middle a...	INUMERIC [SalesData].[Employees].[SAL],SalaryRanges, {Transform definition}, {T...
—	createparamet...		PARAMETRIC [SalesData].[Customers],CustomersQueries, {Transform definition}
—	createquantile		QUANTILE [SalesData].[Customers].[creditLimit],CreditLimitQuant,4, {Transform ...
+	createranking	Rank customers, depending...	RANKING [SalesData].[Customers].[customerNumber],[SalesData].[SaleOrders].[o...

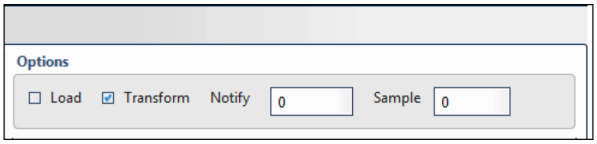
Figure 5-35 Creating a Ranking instruction

11 Save the project.

12 Run the instruction.

1 In Main file, select each instruction and then, choose - to make it inactive, as shown in Figure 5-21.

2 In Data—Options, deselect Load, as shown in Figure 5-36.



Options

□ Load ☑ Transform Notify 0 Sample 0

Figure 5-36 Deselecting Load

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

Managing data objects

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

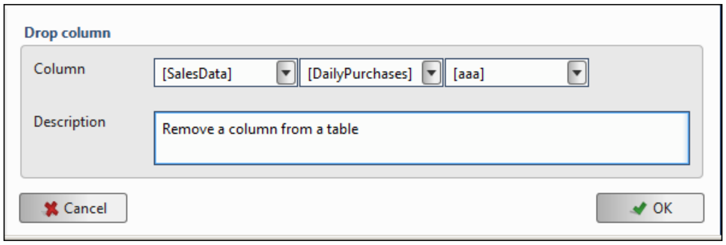
DROPCOLUMN

Removes a column from a table. The syntax is:

Syntax `DROPCOLUMN [Column name]`

Parameter **Column name**

Define the name of the column you want to drop from a table, as shown in Figure 5-37.



Drop column

Column: [SalesData] [DailyPurchases] [aaa]

Description: Remove a column from a table

Figure 5-37 Removing a column from a table

DROPDATABASE

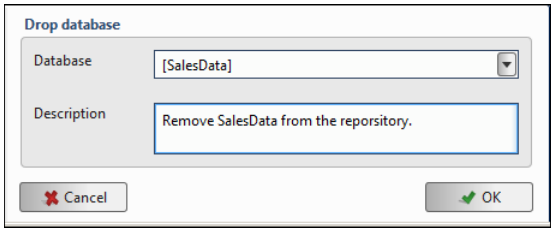
Removes a database from the repository. The syntax is:

Syntax

```
DROPDATABASE [Database name]
```

Parameter

Database name
Defines the name of the database you wish to remove from the repository, as shown in Figure 5-38.



Database: [SalesData]

Description: Remove SalesData from the reporsitory.

Figure 5-38 Removing a database from the repository

DROPTABLE

Removes a table from the repository. The syntax is:

Syntax

```
DROPTABLE [Table name]
```

Parameter

Table name
Defines the name of the table you wish to remove from a database, as shown in Figure 5-39.

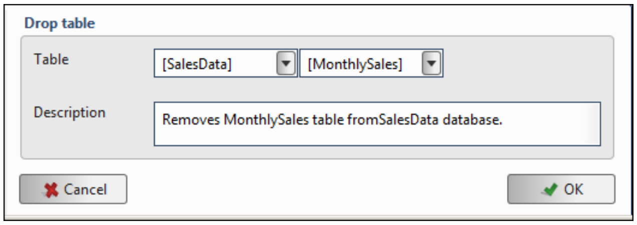


Table: [SalesData] [MonthlySales]

Description: Removes MonthlySales table fromSalesData database.

Figure 5-39 Removing a table from a database

DROPLINK

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

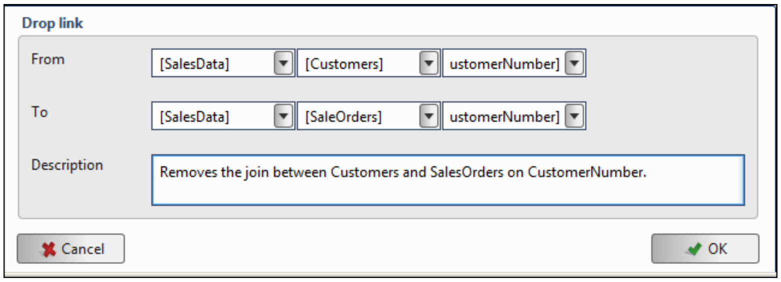
Syntax

```
DROPLINK [From column name] [To column name]
```

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 5-40 shows Drop link.



From: [SalesData] [Customers] ustomerNumber]

To: [SalesData] [SaleOrders] ustomerNumber]

Description: Removes the join between Customers and SalesOrders on CustomerNumber.

Figure 5-40 Removing a link between tables

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. The syntax is:

Syntax

```
LINK [From column name] [To column name]
```

Parameters

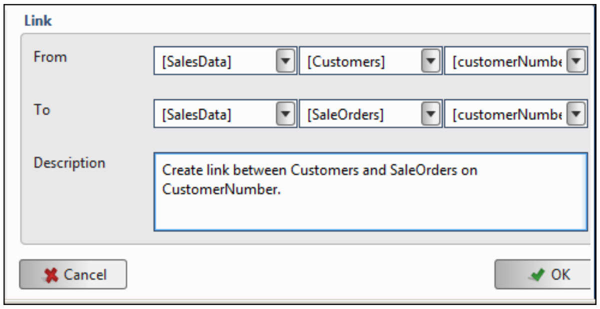
From column name
Defines the name of the column from which the join is initiated.

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 Main file, choose New. The list of transformation instructions appears.
2. In the list of transformation instructions, choose Link. Link appears, as shown in Figure 5-41.



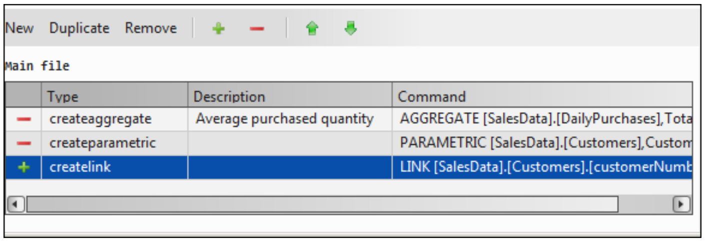
Link

From	[SalesData]	[Customers]	[customerNumbe
To	[SalesData]	[SaleOrders]	[customerNumbe

Description: Create link between Customers and SaleOrders on CustomerNumber.

Figure 5-41 Creating a link between two tables

3. In From, define a database, a table, and a column the link initiates.
4. In To, define a database, a table, and a column to which the link points.
5. In Link, choose OK. The link instruction appears in Main file, as shown in Figure 5-42.



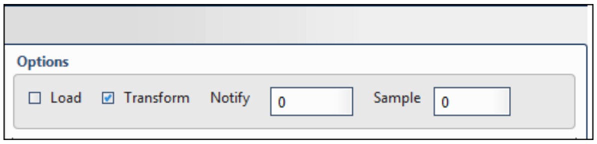
Main file

	Type	Description	Command
−	createaggregate	Average purchased quantity	AGGREGATE [SalesData].[DailyPurchases],Tota
−	createparametric		PARAMETRIC [SalesData].[Customers],Custom
+	createlink		LINK [SalesData].[Customers].[customerNumb

Figure 5-42 Creating a Link instruction

6. Save the project.
7. Run the instruction.

1. In Main file, select each instruction and then, choose - to make it inactive, as shown in Figure 5-21.
2. In Data—Options, deselect Load, as shown in Figure 5-43.


Options

☐ Load ☑ Transform Notify 0 Sample 0

Figure 5-43 Deselecting Load

3 Choose Run. As the project runs, messages describing each stage of the project appear in Running project.

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. The syntax is:

Syntax

```
MAKEPERMANENT [Column name]
```

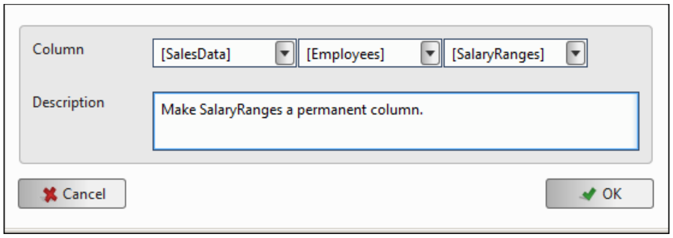
Parameter

Column name

Defines the column name you wish to make permanent, as shown in Figure 5-44. 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.

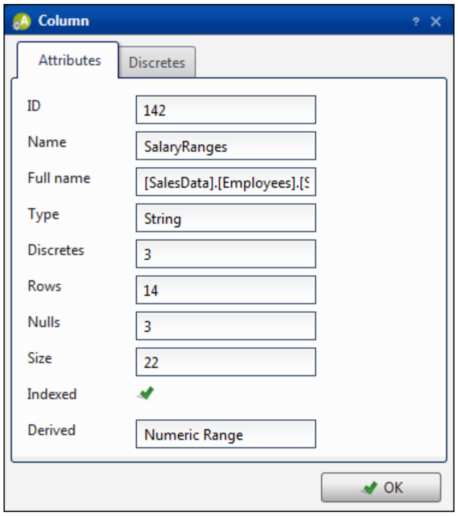

Column [SalesData] [Employees] [SalaryRanges]

Description Make SalaryRanges a permanent column.

Figure 5-44 Making a column permanent

You can validate if the column is converted to permanent in Explorer. Select the column properties and check its Derived property. The property should be Not Derived.

For example, the [SalaryRanges] column is a dynamic column, created using the INUMERIC transformation instruction. The column's Derived property is numeric range before it is made permanent, as shown in Figure 5-45.



ID 142

Name SalaryRanges

Full name [SalesData].[Employees].[S

Type String

Discretes 3

Rows 14

Nulls 3

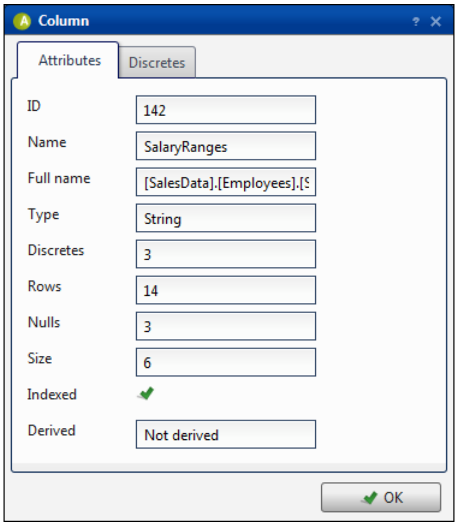
Size 22

Indexed ✓

Derived Numeric Range

Figure 5-45 Dynamic column status

After the transformation the column status is changed to Not derived, as shown in Figure 5-46.



ID	142
Name	SalaryRanges
Full name	[SalesData].[Employees].[S
Type	String
Discretes	3
Rows	14
Nulls	3
Size	6
Indexed	✓
Derived	Not derived

Figure 5-46 Permanent column status

RENAME

Renames a database object, such as a database, a table, or a column. The syntax is:

Syntax

```
RENAME [Old name] [New name]
```

Parameters

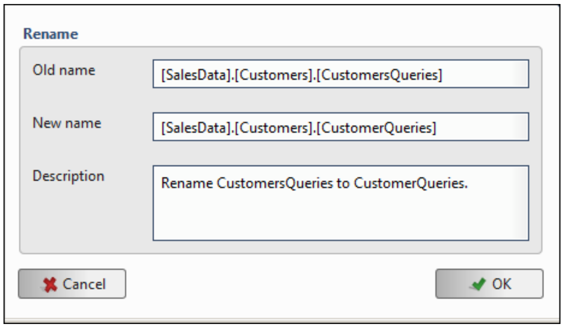
Old name
Defines a current name of a database, a table, or a column in the following format:

```
[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 5-47.



Old name	[SalesData].[Customers].[CustomersQueries]
New name	[SalesData].[Customers].[CustomerQueries]
Description	Rename CustomersQueries to CustomerQueries.

Figure 5-47 Renaming a column

Note, you must synchronize the database and set the privileges after you run the Rename instruction in order to see the database changes.

Miscellaneous instructions

The instructions in this group have different purposes.

EXPORT

Exports a domain to a table or a file. The syntax is:

Syntax

```
EXPORT [Export target] [Target name] [Query resolution table]
   [Query columns] [Filter]
```

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 (with the following delimiters available: coma, colon, semicolon, at, sharp, quote, plus, minus, apos, tilde)
  Defines a CSV file name. Columns are delimited by the pipe (|) character. The file is stored in:

  ```
  C:\Program Files\BIRTAnalytics\data\FastDB
  ```
- Fixed-length file
  Defines a file name. The columns have fixed length and the file is saved in:

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

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]
```

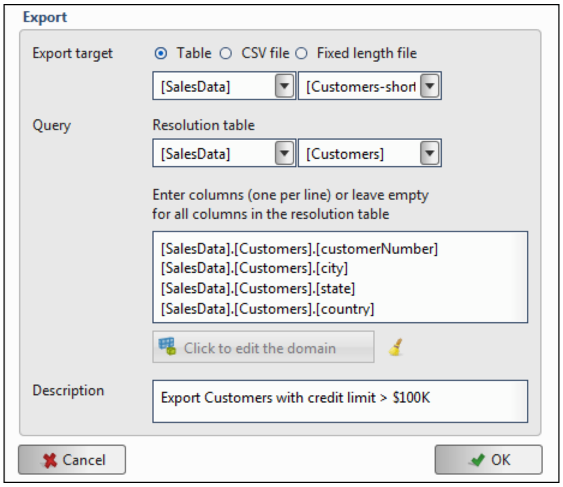
To export the whole table, do not define columns.

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 Main file, choose New. The list of transformation instructions appears.
2. In the list of instructions, choose Export. Export appears, as shown in Figure 5-48.



Export

Export target ● Table ○ CSV file ○ Fixed length file

[SalesData] [Customers-short

Query

Resolution table

[SalesData] [Customers]

Enter columns (one per line) or leave empty for all columns in the resolution table

[SalesData].[Customers].[customerNumber]
[SalesData].[Customers].[city]
[SalesData].[Customers].[state]
[SalesData].[Customers].[country]

Click to edit the domain

Description

Export Customers with credit limit > $100K

Figure 5-48 Exporting data to a table

3. In Export target, select Table.
4. In Export target, type a new database name, or select an existing database name from the drop-down list.
5. In Export target, type a new table name, or select an existing table name from the drop-down list.
6. In Resolution table, select the table from which you want to export data.
7. In Enter columns, type the name of the columns, using the required format. Leave it empty if you wish to export the whole table. You can double-click in the box to open a text editor and type the column list.
8. Choose Click to edit the domain to define a domain that would filter the data. Defining a domain is required.
9. In Description, type a short description of the instruction's purpose.
10. In Export, choose OK to define the Export instruction.
11. Save the project.
12. Run the Export instruction.
13. To validate the export, open Explorer. Refresh the database. The new table appears, as shown in Figure 5-49.

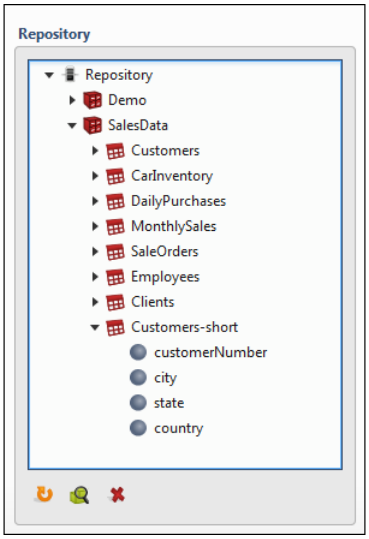
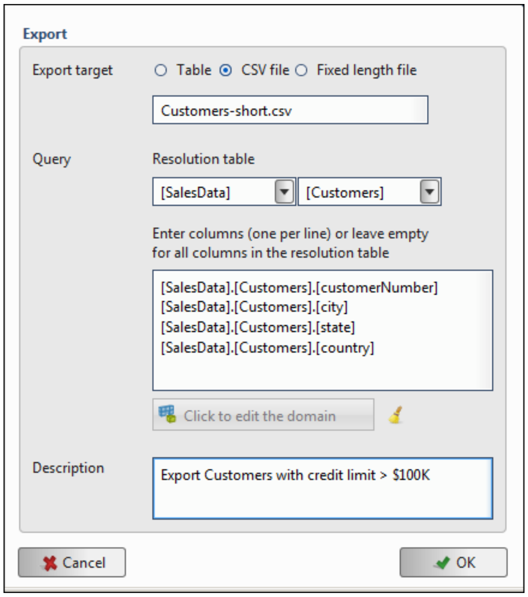


Figure 5-49 Validating an Export instruction

How to export data to a text file

1. In Main file, choose New. The list of transformation instructions appears.
2. In the list of transformation instructions, choose Export. Export appears, as shown in Figure 5-50.
3. In Export target, select CSV file.
4. In Export target, type the name of the CSV file.
5. In Resolution table, select the table from which you want to export data.
6. In Enter columns, type the name of the columns, using the required format. Leave it empty if you wish to export the whole table.
7. Choose Click to edit the domain to define a domain that would filter the data. Defining a domain is required.


Export

Export target ○ Table ● CSV file ○ Fixed length file

Customers-short.csv

Query

Resolution table

[SalesData] [Customers]

Enter columns (one per line) or leave empty for all columns in the resolution table

```
[SalesData].[Customers].[customerNumber]
[SalesData].[Customers].[city]
[SalesData].[Customers].[state]
[SalesData].[Customers].[country]
```

Description: Export Customers with credit limit > $100K

Figure 5-50 Exporting data to a CSV file

8 In Description, type a short description of the instruction's purpose.

9 In Export, choose OK to define the Export instruction.

10 Save the project.

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

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

The file format uses a pipe (|) as a column character delimiter, and CRLF as the end-of-line character, as shown in Listing 5-1.

Listing 5-1 Exported CSV file

```
114|Melbourne|Victoria|Australia
119|Nantes||France
124|San Rafael|CA|USA
131|NYC|NY|USA
141|Madrid||Spain
146|Lyon||France
```

How to export to a fixed-length file

1 In Main File, choose New. The list of transformation instructions appears.

2 In the list of transformation instructions, choose Export. Export appears, as shown in Figure 5-51.

3 In Export target, select Fixed length file.

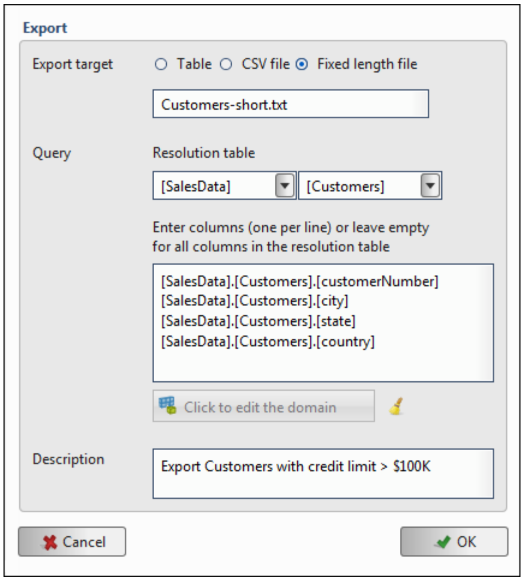
4 In Export, type the name of the fixed-length file.

5 In Resolution table, select the table from which you want to export data.

6 In Enter columns, type the name of the columns, using the required format. Leave it empty if you wish to export the whole table.

7 Choose Click to edit the domain to define a domain that would filter the data. Defining a domain is required.

8 In Description, type a short description of the instruction's purpose.


Export

Export target ○ Table ○ CSV file ● Fixed length file

Customers-short.txt

Query

Resolution table

[SalesData] [Customers]

Enter columns (one per line) or leave empty for all columns in the resolution table

```
[SalesData].[Customers].[customerNumber]
[SalesData].[Customers].[city]
[SalesData].[Customers].[state]
[SalesData].[Customers].[country]
```

Description

Export Customers with credit limit > $100K

Figure 5-51 Exporting data to a fixed-length file

9 In Export, choose OK to define the Export instruction.

10 Save the project.

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

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

The file format uses fixed length for saving the data column, as shown in Listing 5-2.

Listing 5-2 Fixed-length file

```
Melbourne          Victoria                Australia
Nantes                                     France
San Rafael         CA                      USA
NYC                NY                      USA
Madrid                                     Spain
Lyon                                       France
```

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. The syntax is:

Syntax

```
IFEXISTS [Object][Include if true][Include if false]
```

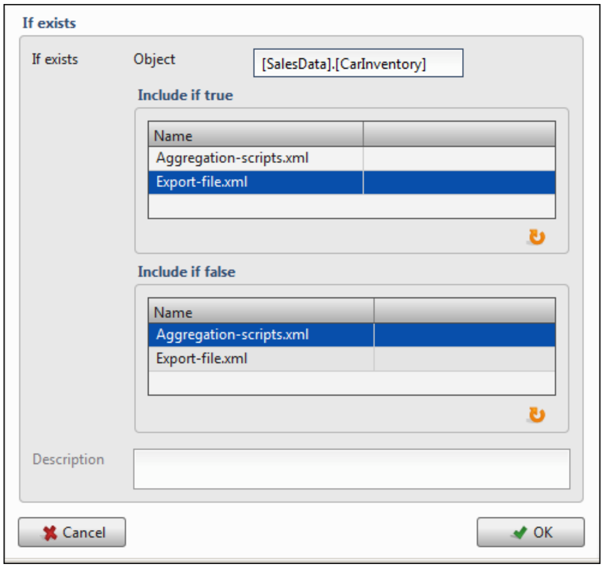
Parameters

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 5-52 checks if the [SalesData].[CarInventory] table exists. The Ifexists instruction executes the Export-file.xml script if the table exists, or Aggregation-scripts.xml, if the table does not exist.



If exists

If exists Object [SalesData].[CarInventory]

Include if true

Name	
Aggregation-scripts.xml	
Export-file.xml	

Include if false

Name	
Aggregation-scripts.xml	
Export-file.xml	

Description

Figure 5-52 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 allows you to dynamically execute scripts, depending on a condition. The syntax is:

Syntax

```
IFLINKED [From link][To link][Include if true][Include if false]
```

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

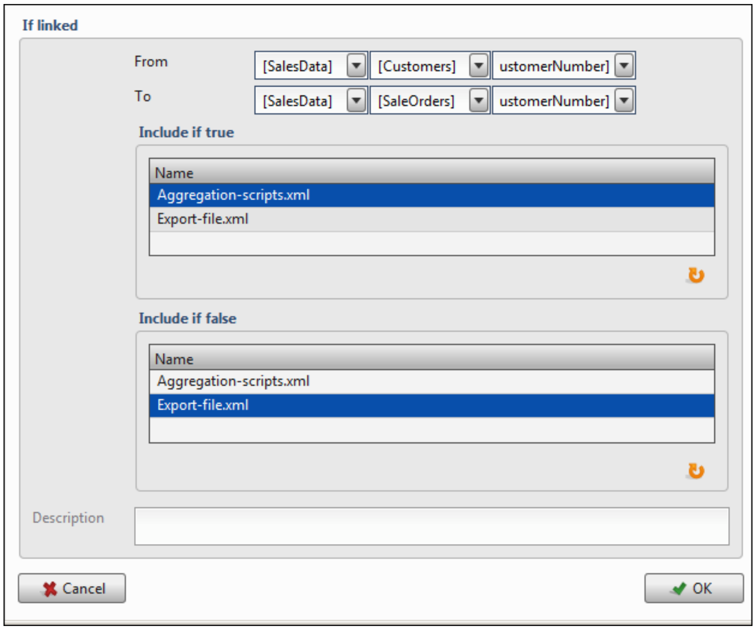
1 In Main file, choose New. In the list of transformation instructions, choose IFLINKED. If linked appears.

2 In From, define a database, a table, and a column from which the link initiates, as shown in Figure 5-53.

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.



If linked

From [SalesData] [Customers] ustomerNumber]

To [SalesData] [SaleOrders] ustomerNumber]

Include if true

Name
Aggregation-scripts.xml
Export-file.xml

Include if false

Name
Aggregation-scripts.xml
Export-file.xml

Description

Figure 5-53 Creating an Iflinked instruction

6 In If linked, choose OK. The Iflinked instruction appears in Main file.

INCLUDE

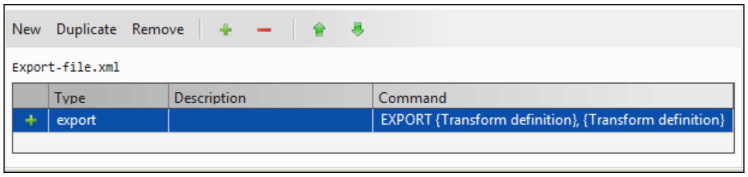
Includes a script file or creates an empty include entry.

Syntax `INCLUDE [File name]`

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, as shown in Figure 5-54, to the main script.



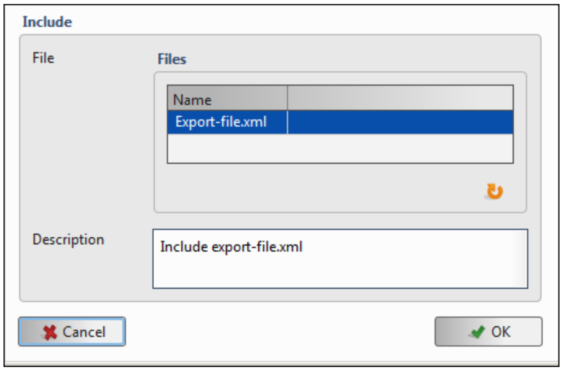
Export-file.xml

Type	Description	Command
export		EXPORT {Transform definition}, {Transform definition}

Figure 5-54 Creating a script file

1 In Main file, choose New. The list of transformation instructions appears.

2 In the list of transformation instructions, choose Include. Include appears, as shown in Figure 5-55.



Export-file.xml

Description: Include export-file.xml

Figure 5-55 Including a script file

3 In Include, click Refresh to make the latest script files appear.

4 In Files, select the file you want to include.

5 In Description, write a short note explaining the instruction.

6 Choose OK. The instruction appears in Main file.

7 Save the project.

PRINT

Writes comments in a project log file. The syntax is:

Syntax

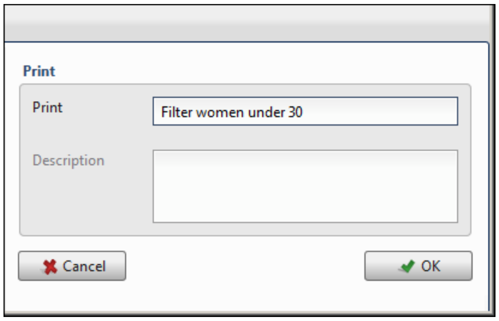
```
PRINT [Text]
```

Parameter

Text
The text to include in the log file.

For example, a comment, like the one shown in Figure 5-56, appears in the log files as:

```
03/05/2013 13:35:37-*INFO-Filter women under 30
```



Print: Filter women under 30

Figure 5-56 Creating a comment

VAR

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

Syntax

```
VAR [Variable name] [Variable value]
```

Parameters

Variable name
The name of the variable. Use descriptive names.

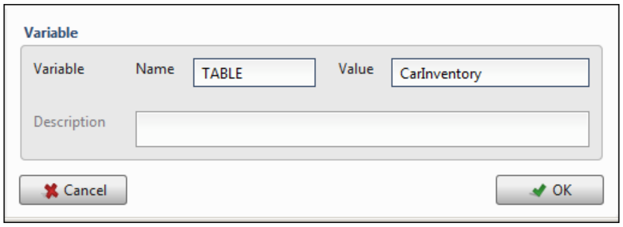
Variable value
The value of the variable. The values can be of any type.

To reference a variable use the following syntax:

```
%VAR%
```

How to use a variable

1. In Main file, choose New. In the list of transformation instructions, choose VAR. Variable appears.
2. In Name, type the variable name, as shown in Figure 5-57. In Value, type the variable value. In this example, the variable contains the name of a table.



Variable

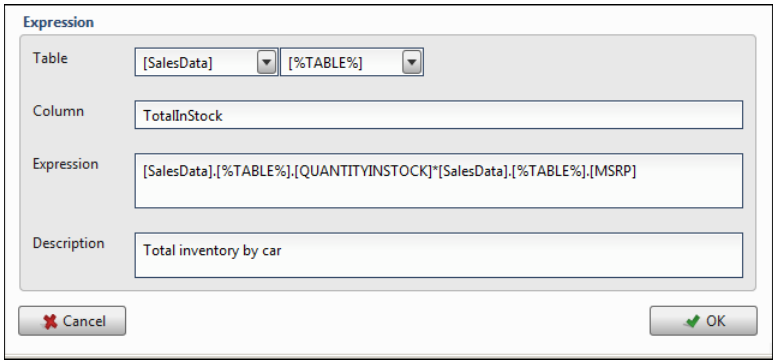
Variable Name: TABLE Value: CarInventory

Description:

Figure 5-57 Creating a variable

3. In Variable, choose OK. Variable appears in Main file.
4. In Main file, choose New. In the list of transformation instructions, choose EXPESSION. Expression appears.
5. In Table, select a database from the drop-down list, and type the following expression in the table box, as shown in Figure 5-58:

   ```
   [%TABLE%]
   ```



Expression

Table: [SalesData] [%TABLE%]

Column: TotalInStock

Expression: [SalesData].[%TABLE%].[QUANTITYINSTOCK]*[SalesData].[%TABLE%].[MSRP]

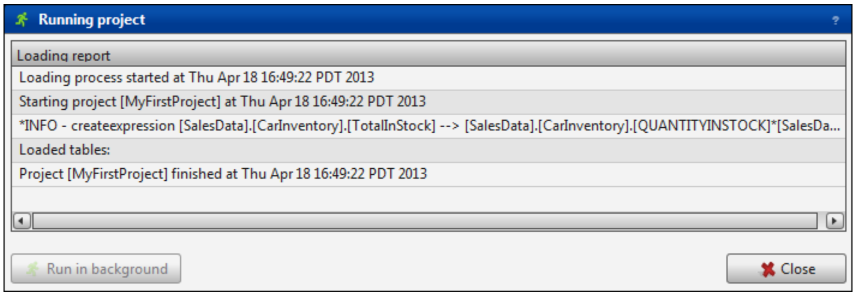
Description: Total inventory by car

Figure 5-58 Using a variable

6. In Column, type the name of the new column.
7. In Expression, type the expression for calculating the value of the new column. Use the variable reference where needed.

   ```
   [SalesData].[%TABLE%].[QUANTITYINSTOCK]*[SalesData].[%TABLE%].[MSRP]
   ```

8. Choose OK, and select Save, to save the expression.
9. Choose Run. The log, as shown in Figure 5-59, displays that the TABLE variable is replaced by its value, CarInventory.



```
Loading report
Loading process started at Thu Apr 18 16:49:22 PDT 2013
Starting project [MyFirstProject] at Thu Apr 18 16:49:22 PDT 2013
*INFO - createexpression [SalesData].[CarInventory].[TotalInStock] --> [SalesData].[CarInventory].[QUANTITYINSTOCK]*[SalesDa...
Loaded tables:
Project [MyFirstProject] finished at Thu Apr 18 16:49:22 PDT 2013
```

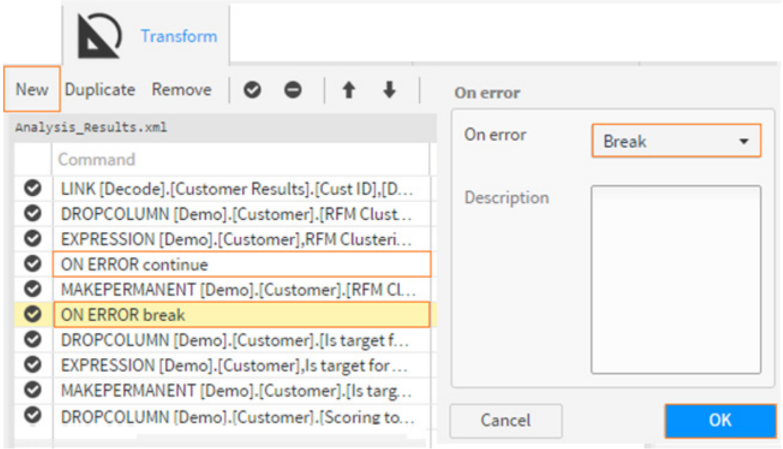
Figure 5-59 Running a project

How to manage the transformation process when exceptions occur

Users can decide *not to stop* the transformation process when errors occur via the TRANSFORM tab. By default, the loading process stops whenever an error is encountered. However, thanks to new *ONERROR instructions* for the transformation script, it is now possible to choose whether you want the transformation process to stop in the case of error or to continue running. This allows the transformation process to ignore any columns that present errors during the process, leaving them to be fixed and processed later if desired.

ON ERROR Instructions

In the TRANSFORM tab, you can create new instructions: *ON ERROR continue* and *ON ERROR break*. This is done by clicking on "New", selecting "ON ERROR" from the list of available instructions that appears and then choosing either "continue" or "break" as needed. Use the "up" and "down" arrows at the top of the tab to position your instructions wherever you want them to be in your rows of instructions. (See Figure 5-60).



Analysis_Results.xml

Command
LINK [Decode].[Customer Results].[Cust ID],[D...
DROPCOLUMN [Demo].[Customer].[RFM Clust...
EXPRESSION [Demo].[Customer],RFM Clusteri...
ON ERROR continue
MAKEPERMANENT [Demo].[Customer].[RFM CL...
ON ERROR break
DROPCOLUMN [Demo].[Customer].[Is target f...
EXPRESSION [Demo].[Customer],Is target for...
MAKEPERMANENT [Demo].[Customer].[Is targ...
DROPCOLUMN [Demo].[Customer].[Scoring to...

On error

On error: Break

Description:

Figure 5-60 Managing exceptions using the Transform tab

Chapter

6

Administering BIRT Loader

This chapter contains:

- About BIRT Loader administration
- Managing users
- Scheduling load processes
- Using External call
- Understanding the BIRT Loader internal database
- Copying a load project to a new location
- Managing the repository

About BIRT Loader administration

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

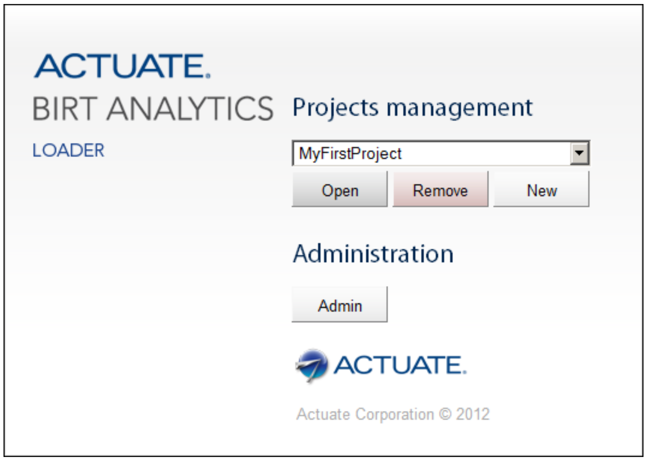
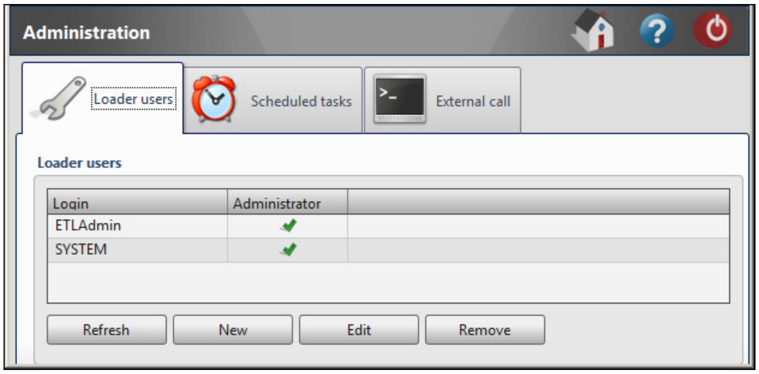


Figure 6-1 Starting Administration

The Administration page of BIRT Analytics Loader is shown in Figure 6-2. The main activities are presented as tabs:

- Loader users
  Manages the BIRT Loader users.
- Scheduled tasks
  Create and manage BIRT Loader scheduled tasks.
- External call
  Returns a credential security file for a selected project. Using the file allows you to run a project remotely.



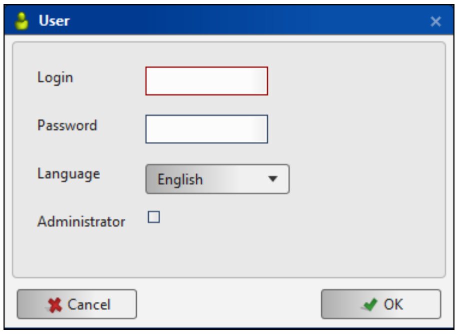
Login	Administrator
ETLAdmin	✓
SYSTEM	✓

Figure 6-2 Administering BIRT Loader

Managing users

Loader users sets up user access to BIRT Analytics Loader. There are three options:

- New
  Create a new BIRT Analytics Loader user, as shown in Figure 6-3.



Login

Password

Language: English

Administrator □

Figure 6-3 Creating a user

    For each user, you must define the following:

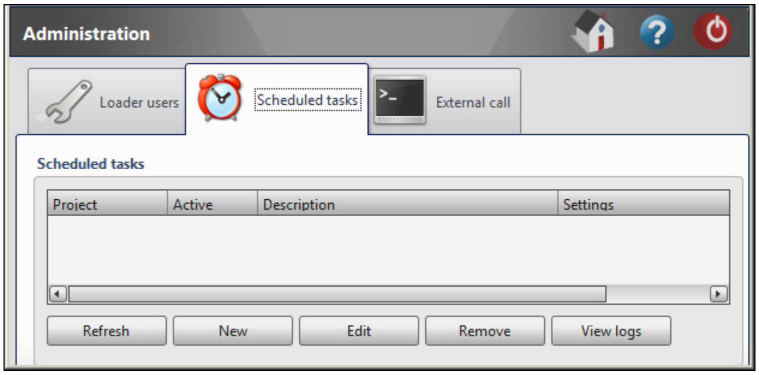
    - Login
      User name.
    - Password
      Password for the user.
    - Language
      The language in which you want the user interface to appear. Currently, only English is supported.
    - Administrator
      Indicate if the user is an administrator.

- Edit
  Edit an existing BIRT Analytics Loader user.
- Remove
  Remove a BIRT Analytics Loader user.

Scheduling load processes

As good practice, to not disturb the analysis process, load data when BIRT Analytics users are not in the system. Scheduling allows you to run the load processes overnight, or at times when the system is idle.

Scheduled tasks allows you to create and manage scheduled tasks, as shown in Figure 6-4.



Scheduled tasks

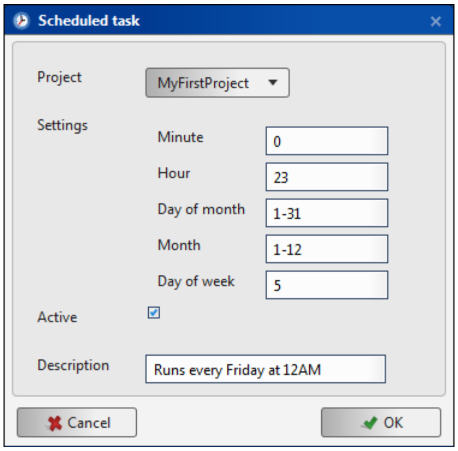
Project	Active	Description	Settings

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:



Scheduled task

Project: MyFirstProject

Settings
- Minute: 0
- Hour: 23
- Day of month: 1-31
- Month: 1-12
- Day of week: 5

Active: ☑

Description: Runs every Friday at 12AM

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.

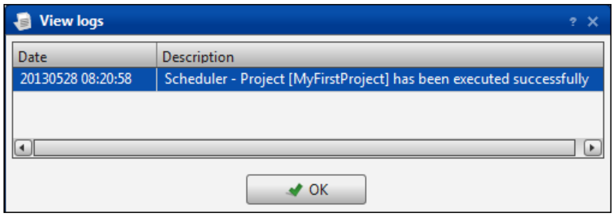
Table 6-1 Scheduling operators

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.

Table 6-1 Scheduling operators

Operators	Description	Examples
*/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.
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 will not run if Active is deselected.
  - 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.

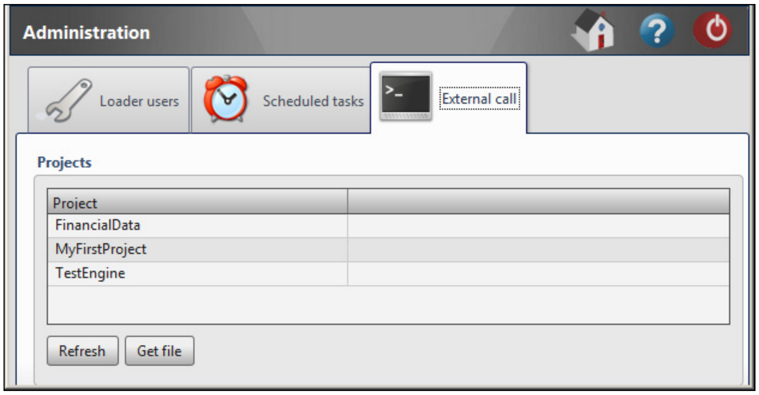


Date	Description
20130528 08:20:58	Scheduler - Project [MyFirstProject] has been executed successfully

Figure 6-6 Viewing scheduled task log

Using External call

External call returns a credential security file for a project, as shown in Figure 6-7. The credential file allows 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.



Project	
FinancialData	
MyFirstProject	
TestEngine	

Figure 6-7 Getting a project credential 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 Loader installer generates and saves in this folder:

```
C:\Program Files\BIRTAnalytics\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 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 the user accesses.
- Get file
  Generates a credential token for the selected project. This token is used to launch 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

qLauncher is a Java-based utility, designed to run 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 1.7, and requires Java 1.7 in order to run.

The qLauncher command syntax is:

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

Parameters

--url
The BIRT 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:8080/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:

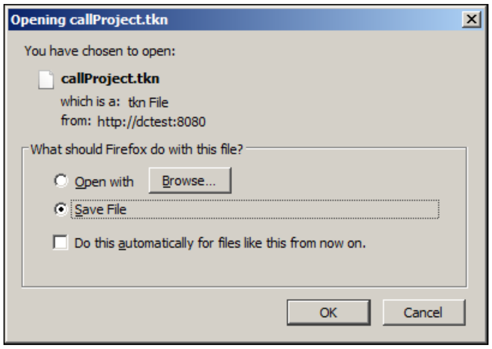
```
java -jar qlauncher.jar
```

When you run a project, the utility changes the public key with the private key to log in to the BIRT 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. Opening callProject.tkn appears, as shown in Figure 6-8.



Opening callProject.tkn

You have chosen to open:

callProject.tkn

which is a: tkn File

from: http://dctest:8080

What should Firefox do with this file?

○ Open with

● Save File

□ Do this automatically for files like this from now on.

Figure 6-8 Opening callProject.tkn

In Opening callProject.tkn, choose OK. 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:

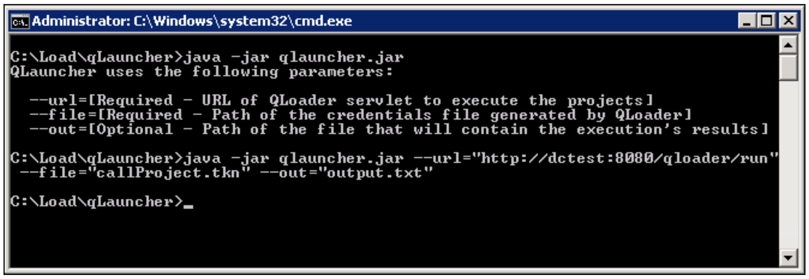
```
java -jar qLauncher.jar
```

The help for the command syntax appears, as shown in Figure 6-9. 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://dctest:8080/qloader/run"
   --file="callProject.tkn" --out="output.txt"
```

The project run finishes and the prompt appears, as shown in Figure 6-9.



Administrator: C:\Windows\system32\cmd.exe

```
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://dctest:8080/qloader/run"
 --file="callProject.tkn" --out="output.txt"

C:\Load\qLauncher>_
```

Figure 6-9 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 Fri Apr 26 14:12:30 PDT 2013
04/26/2013 14:12:31 - Starting project [MyFirstProject] at Fri Apr 26
   14:12:31 PDT 2013
04/26/2013 14:12:31 - *INFO - Table: [SalesData].[Customers]
04/26/2013 14:12:34 - *INFO - Table: [SalesData].[Customers] - 122
04/26/2013 14:12:34 - *INFO - Table: [SalesData].[CarInventory]
04/26/2013 14:12:35 - *INFO - Table: [SalesData].[CarInventory] - 110
04/26/2013 14:12:35 - *INFO - Table: [SalesData].[DailyPurchases]
04/26/2013 14:12:36 - *INFO - Table: [SalesData].[DailyPurchases] - 35
04/26/2013 14:12:36 - *INFO - Table: [SalesData].[MonthlySales]
04/26/2013 14:12:36 - *INFO - Table: [SalesData].[MonthlySales] - 109
04/26/2013 14:12:36 - *INFO - Table: [SalesData].[SaleOrders]
04/26/2013 14:12:38 - *INFO - Table: [SalesData].[SaleOrders] - 326
04/26/2013 14:12:46 - *INFO - Table: [SalesData].[Employees]
04/26/2013 14:12:47 - *INFO - Table: [SalesData].[Employees] - 14
04/26/2013 14:12:48 - *INFO - Table: [SalesData].[Clients]
04/26/2013 14:12:49 - *INFO - Table: [SalesData].[Clients] - 122
04/26/2013 14:12:49 - Loaded tables:
   Customers,CarInventory,DailyPurchases,MonthlySales,SaleOrders,
   Employees,Clients
04/26/2013 14:12:49 - Project [MyFirstProject] finished at Fri Apr 26
   14:12:49 PDT 2013
```

Understanding the BIRT Loader internal database

BIRT Analytics Loader configuration information is stored in dubnium.db, located in:

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

dubnium.db is a sqlite database. To access the sqlite database, you can use SQLite Manager, an add-in in the Firefox web browser. The main tables in dubnium.db are listed in Table 6-2.

Table 6-2 BIRT Loader internal database tables

Table	Description
PROJECTS	Load projects.
BACKUP	Backup of load projects.
USERS	BIRT Analytics Loader users. By default, user Administrator is created.
DATASOURCE	Data sources connection properties.
URIS	Remote data provider URIs.
URIPARAMS	Remote data provider URI parameters.
EXECUTIONS	BIRT Analytics Loader executions.
EXECUTIONLOG	Execution log details.
LANGUAGES	Language details
LOCKS	Users who locked BIRT Analytics Loader.
SHAREDPROJECTS	Backup of load projects.
SHAREDDATASOURCES	Information on shared database connections
SCHEDULER	Information about scheduled tasks.

Table 6-2 BIRT Loader internal database tables

Table	Description
SCHEDULERLOGS	Scheduled tasks logs.
VERSION	Database (dubnium.db) version.

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.

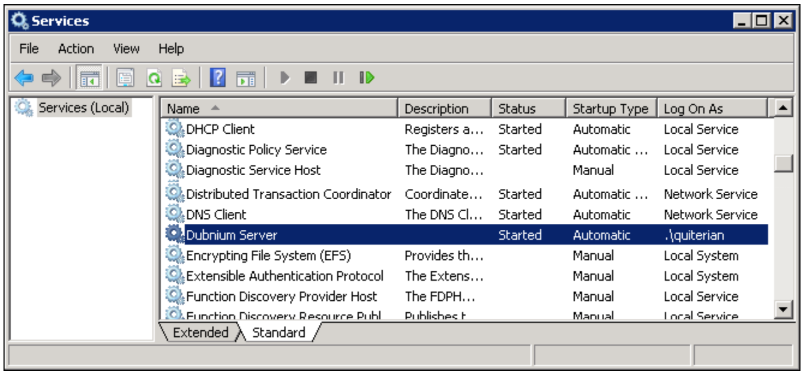
How to copy a load project

1 Stop Dubnium Server.

   1 In Windows➛Start, type:

   ```
   services.msc
   ```

   Services appears, as shown in Figure 6-10.



Name	Description	Status	Startup Type	Log On As
DHCP Client	Registers a…	Started	Automatic	Local Service
Diagnostic Policy Service	The Diagno…	Started	Automatic …	Local Service
Diagnostic Service Host	The Diagno…		Manual	Local Service
Distributed Transaction Coordinator	Coordinate…	Started	Automatic …	Network Service
DNS Client	The DNS Cl…	Started	Automatic	Network Service
Dubnium Server		Started	Automatic	.\quiterian
Encrypting File System (EFS)	Provides th…		Manual	Local System
Extensible Authentication Protocol	The Extens…		Manual	Local System
Function Discovery Provider Host	The FDPH…		Manual	Local Service
Function Discovery Resource Publ	Publishes t		Manual	Local Service

Figure 6-10 Stopping Dubnium Server

2 In Services, select Dubnium Server, and choose Stop from the right-click menu.

2 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"?>
<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>Loader</title>
   <transformationslist>transformations.xml</transformationslist>
   <ignoresslerrors>1</ignoresslerrors>
   <!-- uploading files. Unit is Kb -->
   <maxmemsize>1024</maxmemsize>
   <maxfilesize>51200</maxfilesize>
   <connectionstringsfile>connectionstrings.xml</connectionstringsfile>
</configuration>
```

3 Copy your old project files, except descriptor.txt file, to the new blank project location.

4 Start Dubnium Server.

5 Open the load project from BIRT Analytics Loader and check 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 minimise the required space. Larger repositories require more time to compress and copy. Consider this factor when you plan a backup because your system will not be accessible during the operation.

How to back up a repository

1 Stop Dubnium Server.

2 Make a copy of the current repository content in the following location:

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

3 Start Dubnium Server.

Creating a new database 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 Dubnium Server.

   1 In Windows➛Start, type:

   ```
   services.msc
   ```

   Services appears, as shown in Figure 6-10.

   2 In Services, select Dubnium Server, and choose Stop from the right-click menu.

2 In Windows Explorer, create a new folder where you want to store the Dubnium database. For example:

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

3 Set privileges for the user running the Dubnium server to this folder.

   1 In Windows Explorer, select the folder, and choose Properties from the right-click menu.

   2 In Properties, choose Security.

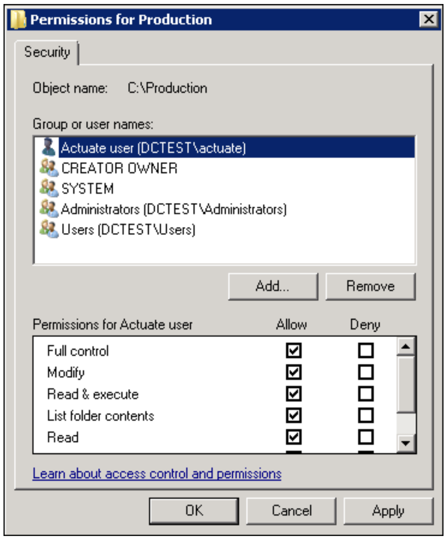
   3 In Security, choose Edit.

   4 In Group or user names, select the user running the Dubnium server, as shown in Figure 6-11. If the user is not in the list, choose Add and select the user to be added.

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

   In Security, choose OK.

   In Properties, choose OK.


Permissions for Production
Security
Add...
Remove
Learn about access control and permissions
OK
Cancel
Apply

Object name: C:\Production

Group or user names:

- Actuate user (DCTEST\actuate)
- CREATOR OWNER
- SYSTEM
- Administrators (DCTEST\Administrators)
- Users (DCTEST\Users)

Permissions for Actuate user	Allow	Deny
Full control	☑	□
Modify	☑	□
Read & execute	☑	□
List folder contents	☑	□
Read	☑	□

Figure 6-11 Setting folder privileges

4 Open Windows Explorer, and navigate to:

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

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]
repository = C:\Program Files\BIRTAnalytics\data\FastDB\databases
[log]
path = C:\Program Files\BIRTAnalytics\log\engine.log
verbosity = 6
[server]
admin_file_path = C:\Program Files\BIRTAnalytics\data\FastDB\admin.sqlite
[memory]
maxmemorysystem = 2048
maxmemorythread = 1024
```

6 Modify the following line:

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

to:

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

7 Save and close the file.

8 In Services, select Dubnium Server and choose Start from the right-click menu.

Copying a data repository to a new location

BIRT Analytics supports changing the location of a data repository by copying the repository to a new location. The Dubnium server must be shut down before copying the repository. Before restarting the Dubnium 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 Dubnium Server.

2 Open Windows Explorer, and navigate to:

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

3 Open the engine_configuration.ini file in Notepad.

4 Modify the following line:

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

for example, to:

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

5 Set privileges for the user running the Dubnium 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 Dubnium Server.

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 check that it works properly.

Changing the repository content using the same location

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 Dubnium Server.

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 Dubnium Server.

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 Loader to verify the restore. Refresh the repository view, if needed.

Adding a BIRT Analytics database to a repository

BIRT Analytics supports copying databases from one repository to another. To do this you must first export the database to a text delimiter file format. The export procedure is described in "EXPORT" in Chapter 5, "Transforming data." Then, import the files to a new database in a new repository using the procedures described in Chapter 3, "Extracting data." Another option for importing data is to use the Import feature in BIRT Analytics. For more details about this option, see *Using BIRT Analytics.*

Glossary

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.

campaign

A set of tasks, defined for specific population segment. A campaign is completed during a defined time period to accomplish a specific goal.

Related term
segment

Canvas

A workspace for data analysis gadgets. Canvas supports arranging, assembling, and saving a collection of data visualization gadgets.

Related term
gadget

cell

A set of properties that defines campaign actions to be performed for all records in a segment.

Related terms
action, campaign, record, segment

Clustering

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.

Related terms
k-means, cluster analysis, predictive analytics

cluster analysis

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.

Related terms
analysis, big data analysis, Clustering

column

1 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.

2 A vertical sequence of cells in a crosstab, grid element, or table element.

Related terms
column-oriented DBMS, database, data field, query, table

column-oriented DBMS

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.

Related terms
database, column

confidence

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:

```
Confidence (A,B-> C) = Support (A,B,C)/Support (A,B)
```

Related terms
association rules, support

consequent

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.

Related terms
antecedent, association rules

Convert

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.

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
A	Calculated	Displays a value result from an expression
10	Date	Contains numbers that represent day, month, and year
10	Date and time	Contains numbers that represent day, month, year, and time of day
1	Full numeric	Contains whole, or integer numbers, such as 1 or 1000
1.0	Real numeric	Contains real, or partial numbers such as 1.05 or 0.003
🕒	Time	Contains a value representing time of day
A	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 (-2,147,483,648) through 2^31-1 (2,147,483,647).
Longint	Integer data from -2^63 (-9,223,372,036,854,775,808) through 2^63-1 (9,223,372,036,854,775,807).
Real	Floating precision number data with the following valid values: -3.40E+38 through -1.18E-38, 0 and 1.18E-38 through 3.40E+38.
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 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 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

Scratchpad A BIRT Analytics work area that supports temporary caching of multiple segments. Scratchpad also supports creating new fields based on segments or selections.

Related terms
Data Explorer, Data Tree, segment, selection

seasonal periodicity

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.

Related terms
Forecasting, Holt-Winters

seasonality In a data set, a periodic trend that corresponds to monthly, quarterly, or semi-annual periods such as seasons.

Related terms
Forecasting, Holt-Winters

security role A set of functionalities that an administrator uses to configure permissions in the BIRT Analytics system.

Related terms
BIRT Analytics Administration, functionalities, query, security filter

security filter A type of query that an administrator uses to limit access to data in the BIRT Analytics system.

Related terms
BIRT Analytics, BIRT Analytics Administration, group, NetScaler Web Logging (NSWL) query

segment A segment is a group of records sharing at least one common characteristic.

Related terms
record, selection

selection A selection is a user-specified request that returns a segment from a database.

Related terms
record, segment

skewness 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.

Related terms
kurtosis, mean

Sphere A data visualization gadget that uses a colored sphere shape to display numeric values and boundaries in a range.

Related terms
Canvas, Cylinder, Dial, Funnel, gadget, Label, Meter

SQL (Structured Query Language)

A language used to access and process data in a relational database.

Related term
database

stage 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.

Related terms
action, campaign, scheduled task, trigger, workflow

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 95
* operator 63, 94
/ (slash) character 95
/ operator 63
+ operator 63
- (hyphen) character 94
- operator 63

A

ABS function 63
access control list 105
access control lists 12
access permissions. *See* privileges
access restrictions 10, 16
accessing
  BIRT Analytics Admin application 13, 92
  BIRT Analytics Loader 3
  data 20
  Dubnium database 98
  projects 8
  text files 26
  Transform text editor 55
accounts 4, 10, 16, 93
ACL to objects option 14
ACL to objects page 14
ACLs. *See* access control lists
actions 32, 105
Active property 95
Actuate BIRT Analytics 2
  See also BIRT Analytics
adding
  aggregate columns 56
  calculated columns 62
  calculated fields 115
  data fields 114
  data sources 21
  database tables 32
  decoding columns 57, 58
  domains 60
  links 49, 79
  load processes 29–30
  parametric columns 71, 72
  quantile range columns 74
  ranking columns 75, 76
  remote data providers 26
  user accounts 16
  users 10, 92
Admin application. *See* BIRT Analytics Administration
Admin module (Loader) 16
Admin page (Analytics Admin) 13
Admin tab (Loader) 10
Admin tools 106
Administration page (Loader) 92
Administration tool. *See* BIRT Analytics Administration
administrator accounts 4
Administrator property 93
Administrator tab (Loader) 4
administrators 3, 6, 92
AGE function 63
Aggregate dialog box 56
aggregate filters 56
AGGREGATE instruction 56
Aggregates tool 105
aggregation functions 56, 57
ALLMONTHS function 63
ALLQUARTERS function 63
analysis (defined) 105
analysis gadgets. *See* gadgets
analytical functions 55
Analytics Admin tool. *See* BIRT Analytics Administration
Analytics application. *See* BIRT Analytics
Analytics Loader. *See* BIRT Analytics Loader
analytics processes 2
  See also ETL processes
analyzing data 2, 10
AND operator 63
antecedent 105
application servers 2
Association Rules 106
association rules 105
Attributes tab (Explorer) 45
authentication 16
automating
  load processes 96
  synchronization processes 16
Average function 56

B

backing up FastDB database 100
Bands parameter 70
baseline filters 106
bashell utility 16
batch processes 95
big data 2
bins 71, 72
Bins parameter 71
BIRT Analytics v, 3
BIRT Analytics Administration 3, 13, 92
BIRT Analytics Administration login page 13
BIRT Analytics Administration tool 106
BIRT Analytics application 106
BIRT Analytics Loader 2, 3, 4, 106
BIRT Analytics Loader command-line tool 16

BIRT Analytics Loader configurations 98
BIRT Analytics Loader engine 96
BIRT Analytics Loader features 9, 10
BIRT Analytics Loader home page 4
BIRT Analytics Loader login page 3
BIRT Analytics Loader port 8
BIRT Analytics Loader user accounts 3
BIRT Analytics Loader user interface 9, 10, 93
Boolean values 63
bubble analysis (defined) 106

C

calculated columns. *See* calculated fields
calculated field type 108
calculated fields 55, 62, 68, 106, 115
calculated values 115
Canvas tool 107
CEIL function 63
changing
  column definitions 37, 38
  data repository 12
  data source definitions 21, 29
  data sources 21
  database names 81
  database records 12
  FastDB database 103
  load processes 29
  load transactions 31
  passwords 10, 17
  project names 9
  projects 9
  remote data providers 26
  scheduled tasks 95
  transformation scripts 53
  user functionalities 15
  user information 93
  user names 4
character sets 33
CHR function 63
closing
  BIRT Analytics Administration 15
  BIRT Analytics Loader sessions 4
  database tables 38
  projects 9
cluster analysis 107
Clustering tool 107
CODE function 63
Code position parameter 58
column bindings. *See* links
Column declaration dialog box 35
column definitions 35, 37, 55
Column name parameter 76, 77, 80
column names 57, 80, 81
column offsets 34
Column to use parameter 76
column transformation instructions 55
column-oriented DBMS 107
columns
  See also calculated fields; fields
  adding to target tables 35
  aggregating values for 56
  changing properties for 37
  changing values in 55
  creating calculated 62, 68
  creating dynamic 47, 55
  creating parametric 71
  decoding 57
  defined 107
  displaying 45, 55
  excluding from load processes 36
  exporting 82
  extracting from source files 34, 39
  grouping values for 69, 74
  linking to 9, 79
  making permanent 47, 80
  ranking record subsets and 75
  removing from tables 36, 77
  renaming 57, 81
  sorting values in 60, 76
  validating 34, 80
  viewing properties for 45, 46, 55
command-line options (qLauncher) 96
command-line tools (BIRT Analytics) 16
comma-separated values files 26, 36
comments (project log files) 88
comments (transformation scripts) 54
computed columns. *See* calculated fields
computed fields. *See* calculated fields
concatenation 62
concentration (values) 112
conditional expressions 63
confidence 107
configuration parameters 9
  See also parameters
configurations 20, 98
connection information 20, 23
connection properties 23, 24
connections
  accessing data and 2, 20, 26
  creating projects and 8
  running native drivers and 23
  running non-native drivers and 24
  setting up 9
  validating 37
consequent 107
constants 63
control menus 9
Convert tool 107
copying
  FastDB database 102
  load projects 99
  repository databases 103
COUNT function 63
Count function 56
count operations 108
Create new project dialog 7

creating
- aggregate columns 56
- calculated columns 62
- calculated fields 115
- data fields 114
- data sources 21
- database tables 32
- decoding columns 57, 58
- domains 60
- dynamic columns 47, 55
- expressions 68
- FastDB database 101
- links 49, 79
- load processes 29–30
- load transactions 30
- mapping specifications 30, 31, 38
- numeric ranges 70
- parametric columns 71, 72
- projects 7–8
- quantile range columns 74
- queries 39, 40
- ranking columns 75, 76
- remote data providers 26
- reports 3
- scheduled tasks 94
- target data tables 35
- transformation script files 53
- transformation scripts 53
- transformation transactions 52
- user accounts 16
- users 10, 92

credential files 95, 96
credential tokens 96
credentials 4, 16, 95
cron 91
Crosstab analysis (defined) 108
CSV files 26, 36
custom data sources. *See* ODBC data sources
customizing Analytics Loader tabs 10
cylinder gadgets 108

D

data
- *See also* values
- accessing 20
- analyzing 2, 10
- controlling access to 10, 16
- grouping 56, 60, 69, 74
- limiting access to 116
- manipulating 52
- protecting sensitive 3, 10, 95
- retrieving 38, 39, 40
- synchronizing 3, 10, 12, 16
- updating 6
- uploading 26, 28
- viewing 9

data analysis 108
data analysis gadgets. *See* gadgets
Data Explorer 108
data export operations 82, 83, 84
data extraction 2, 6, 20, 34
data field names 57, 81
data field types 108
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 44
Data Management tab 9
data mapping models 30, 31
data mining 2
data modeler 6
data object transformation instructions 77
data objects 77
data providers 26, 27
data repository v, 7, 110
- *See also* FastDB database

data rows 45, 46, 115
- *See also* records

data set filters. *See* filters
data sets 38, 39, 40, 71
data source connections. *See* connections
data source definitions 21, 29
data source mapping specifications 9, 20, 30, 31, 38
data sources 2, 6, 21
Data tab (Loader) 9, 20
data transformation 2, 6, 52
data transformation rules 9
Data Tree 109
data types 35, 46, 109
database connection information 20, 23
database connections 20, 23, 24
database drivers 20, 22
database loading processes. *See* load processes
database mapping specifications 38, 39
database model 109
Database name parameter 78
database names 81
database objects 47, 85
database repository. *See* FastDB database
database structures 44
database tables. *See* tables
database types 20
databases 2, 7, 20, 38, 103
- defined 109
- *See also* data sources

Databases panel (Data) 20
Datasource dialog box 23, 24
date and time field type 108
date data type 109
date field type 108
DATE function 63
datetime data type 109
DATETIME function 63

DAY function 64
DAYADD function 64
DAYSTO function 64
decimal values 36, 62
Decision Tree 110
decision tree 109
Decode dialog box 58
DECODE instruction 57
Decode position parameter 58
Decodes tool 110
decoding columns 57
decoding files 58
decryption 96
default data repository 7
deleting
  columns 36, 77
  data sources 21, 26
  links 78
  load transactions 31
  scheduled tasks 95
  script files 53
  tables 78
  transformation instructions 54
  transient links 48, 49
  user accounts 17
  users 93
delimited text files 34
delimiter characters 34, 58
Delimiter parameter 58
Demo database 101
derived column definitions 55
derived columns 55
  See also calculated fields
Description parameter (aggregation) 56
Description property (scheduled tasks) 95
developers v, 6
dial gadgets 110
directory paths 100
discrete values 46, 55
Discrete Values Viewer 110
Discretes tab (Explorer) 46
displaying
  columns 45, 55
  data 9
  database tables 45
  databases 44
  discrete values 46, 55
  load process log files 9, 12, 95
  project log files 12
  project names 8, 9
  script files 88
  transformation instructions 53
  user functionalities 15
DISTINCT function 64
documentation v
Domain definition dialog box 60
DOMAIN instruction 60
domain names 60
Domain parameter 60
domains 60, 82
Downloads tool 110
driver names 24
drivers 2, 20, 22, 24
DROPCOLUMN instruction 77
DROPDATABASE instruction 78
DROPLINK instruction 78
DROPTABLE instruction 78
Dubnium database 4, 98
Dubnium servers 99, 101
Dubnium service 2
Dubnium service security accounts 3, 16
dubnium.exe 110
dynamic columns 47, 55, 80

E

edit panel (table declarations) 39
editable columns 47
editor (Transform) 55
Encoding parameter 58
encoding types 33
encryption 96
End line parameter 58
end-of-line characters 34, 58
EQ operator 63
estimation method 111
ETL (defined) 2
ETL developers v, 6
ETL processes 2, 6
  See also specific process
events 105
evolution analysis (defined) 110
EXP function 64
Explorer 44–47
Export Analytic DB tool 110
Export dialog box 82, 83, 84
Export File tool 110
EXPORT instruction 82
export operations 82, 83, 84
Export target and name parameter 82
export tools 110
Expression dialog box 68
EXPRESSION instruction 62
Expression parameter 63
expressions 62, 68
  defined 110
External Call page (Administration) 95
extraction processes 2, 6, 20, 34

F

FastDB database
  accessing files for 99
  adding databases to 103
  backing up 100
  changing 12, 103
  copying 102
  creating 101

installing 100
loading data into 7
removing databases from 78
removing objects from 47
renaming data fields in 57, 81
renaming databases in 81
running 2
setting up security accounts for 16
testing for objects in 85
viewing database structures in 44
FastDB engine 2
FastDB engine user accounts 3, 10, 16
FastDB web service 111
features 9, 10
field (defined) 108
field names 57, 81
field types 108
fields
adding calculated 115
building summary tables and 112
creating 114
See also columns
file connections 26
file encoding types 33
file headers 34
file lists 28
file names 29
File parameter 58
file paths 100
file systems 27
file types 33
File Upload dialog box 27
files
accessing data in 26
copying project 99
creating script 53
data extraction and multiple 28
defining mapping specifications for 31, 36
exporting data to 82, 83, 84
generating credential 95
opening script 53
selecting formatting properties for 33
transforming data and 52
uploading 27, 28, 58
viewing log 9, 12, 95
Files panel (Data) 20, 26
Filter parameter
AGGREGATE instruction 56
EXPORT instruction 82
INUMERIC instruction 70
QUANTILE instruction 74
RANKING instruction 76
filter types 111
See also specific type
filtering
record groups 106
filters 111
final links 48
First function 56
flat file data sources 26
flat files 20, 34
See also data sources
FLOOR function 64
Force sequence parameter 76
Forecasting tool 111
formatting properties (source files) 33
formula names 47
formulas 47, 55
See also expressions
full numeric field type 108
See also numeric fields
Function parameter 56
functionalities 15, 111
functions 55, 63
funnel gadgets 111

G

gadget types 111
gadgets 111
Gallery 111
GE operator 63
geographic maps 113
grouping
records 116
grouping data 56, 60, 69, 74
groups (users) 12, 111
GT operator 63

H

has seasonality option 111
help (qLauncher) 96
Holt-Winters method 111
home page (Loader) 4
HOUR function 64

I

IF function 64
If statements 63
IFEXISTS instruction 85
IFLINKED instruction 86
IIS services 10
import operations 112
importing metadata 37
inactive transformation instructions 71
Include dialog box 87
INCLUDE instruction 87
INCLUDE statements 85, 87
Includes panel (Transform) 53
indexed fields 112
See also summary tables
indexes 34, 46
information. *See* data
INSTR function 64
INT function 64
integer data type 109

Internet Information Services. *See* IIS services
INUMERIC instruction 69
inversion 60
IP addresses 3, 8
IP parameter 8
ISNOTNULL function 64
ISNULL function 64

K

key pairs (encryption) 96
Key to rank parameter 76
KURT function 64
Kurtosis 112
Kurtosis coefficient 112

L

label gadgets 112
Language property 93
Last function 56
LCASE function 64
LE operator 63
LEFT function 64
LEN function 64
leverage 112
lift 112
Link dialog box 79
LINK instruction 79
links 78, 79, 86
Links tool 44, 48–49, 112
load operation types 32
load processes
  automating 96
  building mapping models for 30, 31
  building projects for 3, 6, 7
  configuring 20
  creating 29–30
  defining connections for 20, 23, 26
  excluding transactions from 32
  running 2, 9
  saving changes to 29
  scheduling 4, 93
  setting options for 30
  starting 10
  viewing log files for 9, 95
load projects. *See* projects
load transactions
  creating 30
  defining source file formats for 33, 36
  defining table columns for 34
  defining target objects for 32, 38, 39
  excluding from load processes 32
  loading data and 20, 29, 40
  running 30
Loader module 106
Loader module. *See* BIRT Analytics Loader
loading processes. *See* load processes
local file systems 27
log files 9, 12, 88, 95
LOG function 64
LOG10 function 64
logging load operations 30
login credentials 4
login names 93
login page (Analytics Administration) 13
login page (Analytics Loader) 3
Login property 93
Logout icon (Loader) 4, 9
Longint data type 109
LONGINT function 64
LT operator 63
LTRIM function 65

M

Main file link 52, 53
make permanent operations 113
MAKEPERMANENT instruction 80
map analysis (defined) 113
mapping 30
mapping specifications 9, 20, 30, 31, 38
mathematical operators 63
MAX function 65
Maximum value function 56
maximum values 113
MEAN function 65
Mean function 56
mean values 113
MEDIAN function 65
median values 113
menus 9
metadata 37
meter gadgets 113
Microsoft Internet Information Services. *See* IIS services
Microsoft SQL Server databases 20
MID function 65
MIN function 65
Minimum value function 56
minimum values 113
MINUTE function 65
MODE function 65
mode values 113
MONTH function 65
My Data Viewer 113
My Folders Viewer 113
MySQL databases 20, 22

N

Name parameter 8
naming
  database connections 23, 24
  decoding columns 59
  domains 60
  FastDB data fields 57, 81
  file data sources 29
  ODBC database connections 25
  parametric columns 72

projects 8
quantile range columns 75
ranking columns 76
variables 88
native drivers 20, 22
NE operator 63
negative values 116
NetScaler Web Logging queries 114
New file list dialog box 28
non-native drivers 21
normal distribution 114
NOT operator 63
NOW function 65
NSWL queries 114
null values 46
Number of parts parameter 74
numeric field types 108
numeric ranges 69, 70
Numeric ranges dialog box 70
Numeric Ranges tool 114
numeric values 69, 74
numerical estimation method. *See* Holt-Winters method

O

ODBC connectors 2
ODBC data sources 24
ODBC driver names 24
ODBC drivers 21, 24
online documentation v
opening
BIRT Analytics Admin application 13, 92
BIRT Analytics Loader 3
BIRT Analytics Loader projects 8
Dubnium database 98
script files 53
Transform text editor 55
operators 63, 94
Options panel (Data) 20, 30
OR operator 63
Oracle databases 20, 39
output files (qLauncher) 98

P

parameters
adding decoding columns and 57
adding parametric columns and 71
adding quantile range columns and 74
adding ranking columns and 76
aggregating data values and 56
changing project configurations and 9
creating domains and 60
creating expressions and 62
creating projects and 8
defined 114
defining numeric ranges and 69
defining variables and 88
dropping database objects and 77, 78
dropping links and 78
entering values for 115
exporting domains and 82
linking tables and 79
opening Loader module and 3
running INCLUDE statements and 85
running qLauncher utility and 96
testing for links and 86
uploading remote files and 27
parametric columns 71
Parametric dialog box 72
parametric filters 114
PARAMETRIC instruction 71
Parametric tool 114
Pareto analysis (defined) 114
Pareto principle 114
Password parameter 8
Password property 93
passwords 4, 10, 17, 93
paths 100
periodic trends 116
permanent columns 47, 80
permissions. *See* privileges; functionalities
Port parameter 8
positive values 116
PostgreSQL databases 21
POWER function 65
predictive analytics 114
private encryption keys 96
privileges 10, 12, 101, 105, 111
processes. *See* ETL processes
profile (users) 115
profile analysis (defined) 114
project development page 9
project files 99
project folders 99
project log files 12, 88
project names 8, 9
Project property 94
projects
building ETL processes and 3, 6
creating 7
limiting log entries for 11
managing 9
naming 8
running 11, 96, 97
saving 9, 38
storing 99
Projects management page 8, 92
prompted filter 115
PROPER function 65
PROPERSENTENCE function 65
protecting data 3, 10, 95
public encryption keys 96
public-key encryption 96

Q

qLauncher utility 96, 97

qualifier characters 34, 58
Qualifier parameter 58
Quantile dialog box 74
QUANTILE instruction 74
quantile range columns 74
quantile range filters 74
Quantile tool 115
QUARTER function 65
queries
  creating 39, 40
  defined 115
  defining bins and 72
  defining domains and 60, 61
  limiting data access and 116
  retrieving data and 38, 39, 71
  retrieving data with 115
  tracking HTTP data traffic and 114
Query columns parameter 82
query language 116
Query resolution table parameter 82

R

RAND function 65
range of values 69, 70
ranking 75
ranking columns 75
Ranking dialog box 76
RANKING instruction 76
ranking keys 76
ranking sequence 76
Ranking tool 115
real data type 109
REAL function 65
real numeric field type 108
  See also numeric fields
records
  adding to tables 32
  changing 12
  creating sample of 60
  defined 115
  grouping 60, 116
  linking tables and 48
  ranking subsets of 75
  setting number of 30
  updating 12
  validating content 34
relational databases 20
remote data providers 26, 27
removing
  columns 36, 77
  data sources 21, 26
  links 78
  load transactions 31
  scheduled tasks 95
  script files 53
  tables 78
  transformation instructions 54
  transient links 48, 49
  user accounts 17
  users 93
RENAME instruction 81
renaming
  databases 81
  FastDB database fields 57, 81
  projects 9
  SYSTEM user 4
REPLACE function 65
REPORT instruction 88
reports 3, 12
repository v, 7, 110
  See also FastDB database
Repository list (Transform) 55
Repository tree 44, 45
resolution 60, 78
RIGHT function 66
roles 116, 118
ROW function 66
rows 45, 46, 115
  See also records
RTRIM function 66
rules (data transformation) 9
running
  Analytics Loader projects 11, 12, 95, 96, 97
  load processes 2, 9
  load transactions 30
  transformation instructions 54, 71
  transformation scripts 53, 86, 88
Running project log 11

S

saving
  load processes 29
  projects 9, 38
Scheduled task dialog 94
scheduled tasks 115
Scheduled tasks page (Administration) 93
scheduling load processes 4, 93
Scratchpad 116
script files 52, 53, 87
scripts. *See* transformation scripts
seasonal patterns 116
seasonal periodicity 116
seasonal trends 111
seasonality 116
SECOND function 66
SECSTO function 66
secure connections 8
security 3, 10, 95
security accounts (Dubnium service) 3, 16
security filters 116
security groups 12
security keys 96
Security module (Analytics Admin) 13
Security page (Analytics Admin) 13
security roles 116
segment blocks 60

segments 60
  defined 116
selections 116
server names 3
servers 2, 10
Settings property 94
Settings tab (Loader) 9
SGN function 66
sharing connections 20
SKEW function 66
skewness operations 116
skewness values 116
Sorting direction parameter 76
Source column parameter 56, 57, 69, 74
source files 33, 34
sphere gadgets 116
spreadsheets 26
SQL (defined) 116
SQL queries. *See* queries
SQL Server databases 20
sqlite database 4, 98
SQLite Manager 98
SQRT function 66
Square sum function 56
squared values 117
SSL parameter 8
stage 116
standard deviation 117
Standard deviation function 56
standard normal distribution 117
Standardize column tool 117
starting
  load processes 10
  synchronization processes 12, 16
STDEV function 66
stopping
  Dubnium servers 99
string data type 109
STRING function 66
STRIP function 66
sum (defined) 117
SUM function 66
Sum function 56
summary values 117
sum-of-squares (defined) 117
SUMSQ function 66
support 117
symmetric distribution 116
Synchronize option 13
Synchronize page (Analytics Admin) 14
synchronizing data 3, 10, 12, 16
system administration tool. *See* BIRT Analytics Administration
system administrators. *See* administrators
system privileges 111
SYSTEM user 4

T

Table declaration dialog box 32, 35, 38
table definitions 6, 9, 21, 29, 32
Table name parameter 78
tables
  adding calculated columns to 55
  adding dynamic columns to 55
  adding fields to 114
  binding columns to. *See* links
  closing 38
  creating 32
  defined 117
  deleting 78
  exporting 82
  grouping data from multiple 56
  grouping records in. *See* segments
  linking to 44, 48–49, 78, 79
  mapping source file data and 31
  ranking 75
  removing columns in 36, 77
  renaming 81
  selecting 39
  setting sorting direction for 76
  testing for links in 86
  updating 6, 12
  viewing columns in 45, 55
  viewing list of 45
  viewing properties for 45
tabs 9, 10
Target column parameter
  AGGREGATE instruction 56
  DECODE instruction 57
  DOMAIN instruction 60
  EXPRESSION instruction 62
  INUMERIC instruction 69
  PARAMETRIC instruction 71
  QUANTILE instruction 74
target filters 117
target objects 30, 32, 39
Target table parameter
  AGGREGATE instruction 56
  DOMAIN instruction 60
  EXPRESSION instruction 62
  PARAMETRIC instruction 71
temporal files 117
temporary files 117
text editor (Transform) 55
text field type 108
text file data sources 26
text files 34, 83
  See also data sources
Text parameter 88
time data type 109
time field type 108
TIME function 66
time settings (scheduled tasks) 94
TODAY function 66
tokens (credentials) 96

tools menu 10
transactions 52
    See also load transactions
Transform tab (Loader) 9, 52
transformation instructions 53, 71, 81
    See also transformation scripts
transformation processes 2, 6, 52
transformation rules 9
transformation script files 52, 53, 87
transformation scripts 30, 52, 53, 86, 88
transformation transactions 52
transient files 117
transient links 48
triggers 117
TRIM function 67

U

UCASE function 67
Unicode data type 109
universal filters 118
updating
    database table definitions 21, 29
    database tables 6, 12
    Includes file list 53
uploading
    data 26, 28
    decoding files 58
    local files 27
    multiple files 28
    remote files 27
URI dialog box 27
URI property 27
URIs 27
URLs
    BIRT Analytics Administration 13
    BIRT Analytics Loader 3
user accounts 4, 10, 16, 93
user credentials 4, 16, 95
user groups 12, 111
user interface 9, 10, 93
user names 4
User parameter 8
user privileges 10, 12, 101
users 3, 10, 92
user-specified requests. *See* selections

V

VAL function 67
values
    aggregating 56
    defining range of 69, 70
    getting from multiple tables 71
    grouping 69, 74
    inverting 60
    loading 47
    ranking 115
    viewing discrete 46, 55
    viewing number of discrete 46
    viewing number of null 46
VAR instruction 88
Variable dialog box 89
variables 88
variance 118
Venn analysis (defined) 118
viewers 113
viewing
    columns 45, 55
    data 9
    database tables 45
    databases 44
    discrete values 46, 55
    load process log files 9, 12, 95
    project log files 12
    project names 8, 9
    script files 88
    transformation instructions 53
    user functionalities 15

W

web addresses 27
web browsers 3, 10
web servers 10
web services v
web services. *See* BIRT Analytics application; FastDB web service
WEEKDAY function 67
workflow 118
workflow roles 118

Y

YEAR function 67

Z

zero values 116
Z-score operations 118