OZ Designer

User Guide for OZ Query Designer

FORCS

Table of Contents

Welcome to Experience OZ Query Designer.

With OZ Query Designer, user can connect to not only the commercial databases like Oracle and MSSQL, but also diversity of data sources like TXT and XML file, and create queries and datasets.

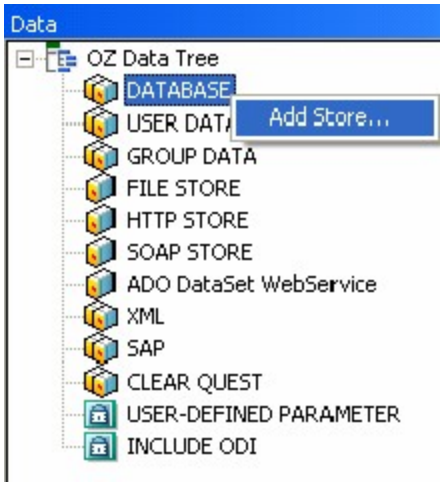
OZ Query Designer was awarded a GS(Good Software) certification, VeriTest certification, SAP NetWeaver certification, CA Smart certification, and chosen as the administrative office software, and won the Korea Grand Prix of the new software sector. OZ Query Designer is worthwhile product by its reliability and usefulness.

OZ Query Designer design or define the dataset to be used by OZ report and user can quickly and easily create professional and complex query. OZ Query Designer helps you to improve development productivity and efficiency of management and maintenance.

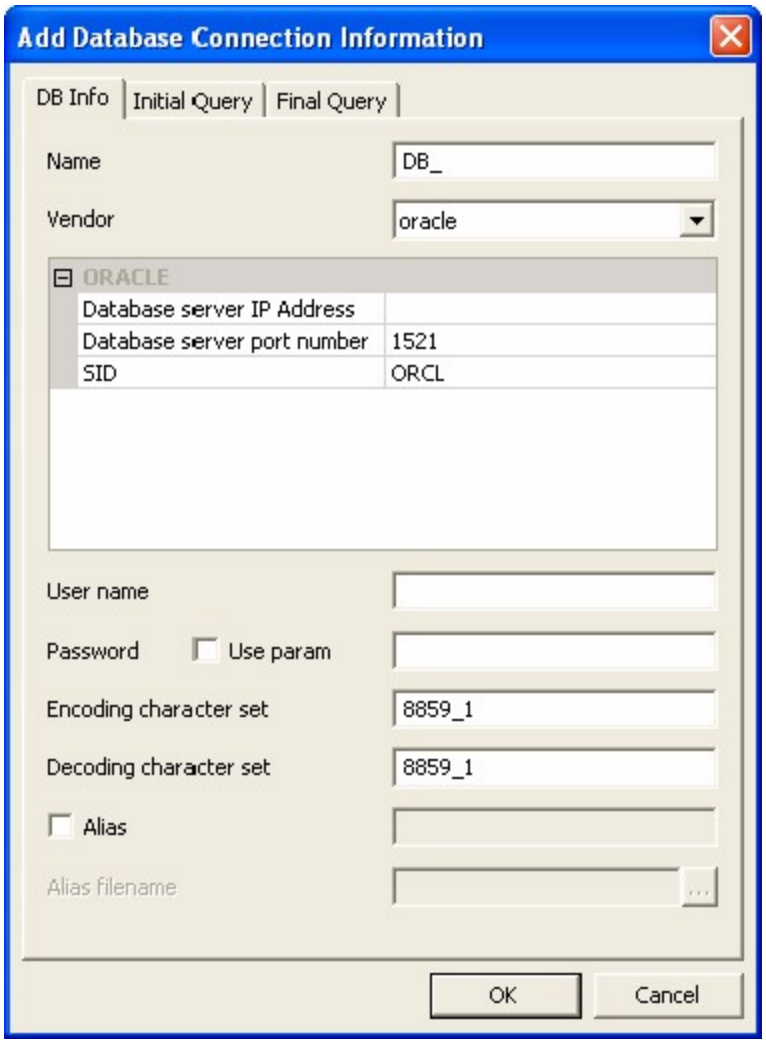
This chapter describes how to create store and dataset.

Using database store, the user can creates SQL and dataset from database.

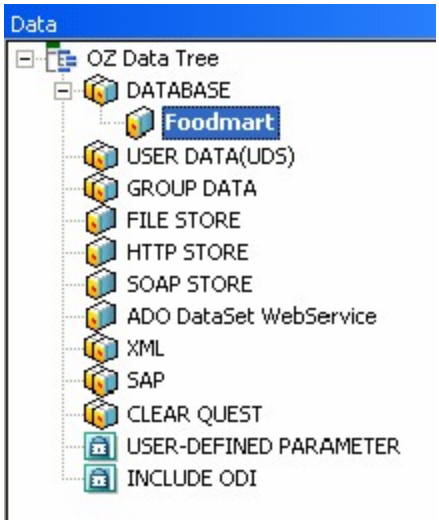
Click the right mouse button on the [DATABASE] item and click **[Add Store...]** again.



Set all needed options in [Add Database Connection Information] dialog and click **[OK]**.



Your database connection appears under [DATABASE] item.



With [Add Database Connection Information] dialog, you can establish database connection.

To run a [Add Database Connection Information] dialog, select **[Add Store...]** from the right mouse click popup menu on the DATABASE in OZ data window.
To establish database connection using [Add Database Connection Information] dialog, enter the vendor name, database server IP address, port number, user name and password.

ODBC connection

Various databases providing ODBC driver such as MS Access can be connected by ODBC connection. Correct ODBC driver should be installed and your database should be registered as DSN(data source name) using the ODBC Data Source Administrator dialog of Windows system.

To establish ODBC database connection, enter your string for "Name" - the database connection ID, and select **odbc** for "Vendor", and enter the DSN registered in Windows for "Data Source Name".

DBMS connection

Set connection to database such as MS-SQL, Oracle, Sybase, Informix, Uni-SQL, DB2, tibero, tibero5, mariadb, etc.

Enter your string for "Name" - the database connection ID, and select your database for "Vendor", and enter Database server IP Address, Database server port number, Database server name, Database name.

Enter database user name and password. If check "Use param" option, password value comes from a user defined parameter.

When connect to sybase, Enter "Tds" for "Data source name" to use jConnect - JDBC driver for sybase. Tds(Tabular Data Stream) is a native communication protocol for sybase Adative Server.

OLEDB connection

Set connection to OLEDB such as MSSQL-OLEDB, ORACLE-OLEDB, SYBASE-OLEDB, DB2-OLEDB, INFORMIX-OLEDB, etc.

Enter your string for "Name" - the database connection ID, and select your oledb(mssql-oledb, oracle-oledb, sybase-oledb, db2-oledb, informix-oledb, etc.) for "Vendor", and enter Database server IP Address, Database server port number, Database name, and Data source name, etc.

Enter database user name and password. If check "Use param" option, password value comes from a user defined parameter.

User-Definded connection

If your database is not in the built-in DB list, you can set your JDBC driver by yourself.

Enter your string for "Name" - the database connection ID, and select "user" for "Vendor", and enter Driver Class Name of your JDBC driver and Connection URL.

Enter database user name and password. If check "Use param" option, password value comes from a user defined parameter.

You can set DB connection using alias instead of entering server IP address and port number using [Add Database Connection Information] dialog. First, Set DB connections information for

each database in db.properties file. Next, open [Add Database Connection Information] dialog and check "Alias" option and set alias file path and alias name. If use alias connection, you don't need to edit ODI file every time when connection information of the database server has changed.

db.properties file example on MS-SQL database connection:

```
Sample.vendor=mssql
Sample.serverAddress=127.0.0.1
Sample.portNo=1433
Sample.dbName=Foodmart
Sample.user=user1
Sample.password=user1password
Sample.maxconns=20
Sample.initconns=5
Sample.timeout=5
```

Check the "Alias" in the [Add Database Connection Information] dialog. Then enter the alias name set in db.properties file and set the path of db.properties file.

Database connection information in db.properties file depends on database vendor such as ODBC, MS-SQL, DB2, Oracle, etc. Sample db.properties file that including needed options for each vendor can be found under /conf/sample relative to OZ installation path. For detailed options, refer to OZ Enterprise Server Administrator's Guide.

It is recommended that path of the alias information file is set as relative path rather than absolute path. For setting a relative path, refer to "Reference > File path setup > Entering path > Relative path"

With user defined parameter, database connection options or database alias name can be set dynamically at runtime.

Connection option setup

Click **[Add Parameter Field]** in popup menu on the USER-DEFINED PARAMETER item in OZ data window.
[USER-DEFINED PARAMETER] dialog appears. Register user defined parameter for each options(IP, PortNo, Username, Password) and set their properties (Name, Type, Value).

Database connection will be made using parameter values passed from report application at runtime. Dynamic database connection with parameter may not protect database connection information, so it is recommended to use dynamic alias name with db.properties file.

Set Initial SQL to be executed immediately after DB connection and Final SQL to be executed immediately before DB disconnection.
Check the "Set query name" in the [Add Database Connection Information] dialog and enter the name. Then enter a Query statement and click the [Add] button.
To use the script query, Check the "Use Scripted Query".

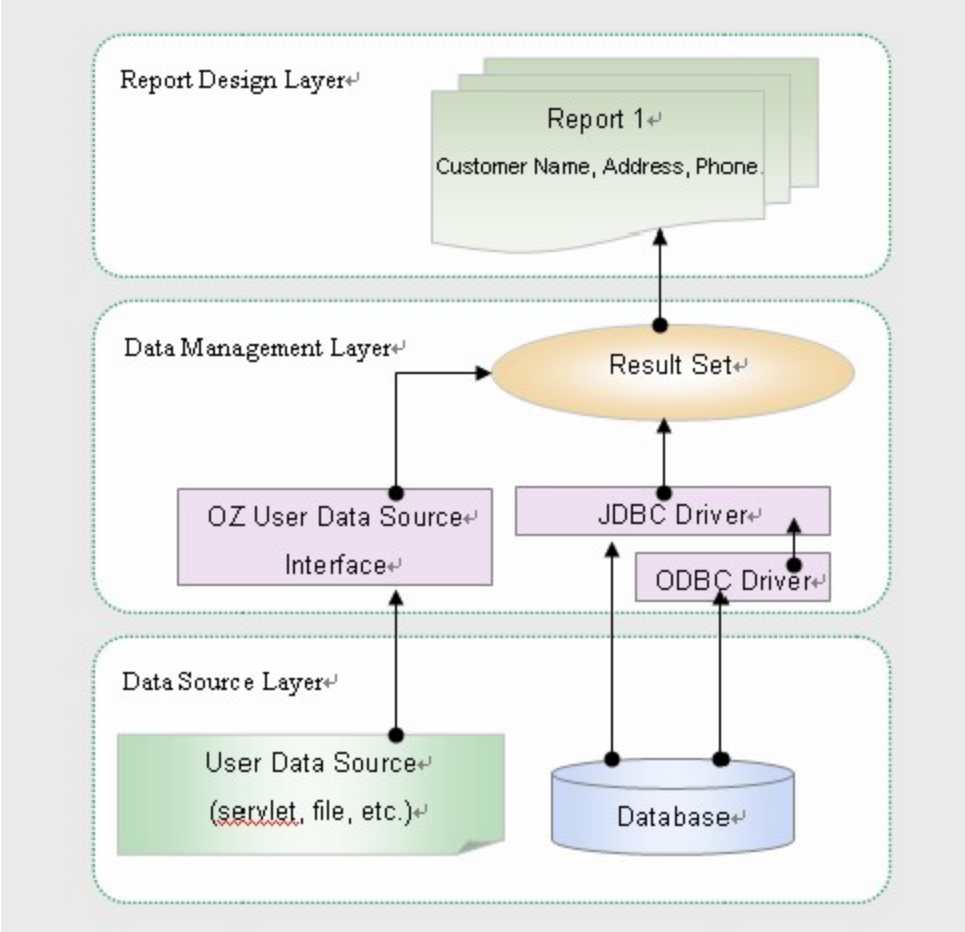
Query statement must be the standard SQL statement can be executed in JDBC. Also It must be a SQL statement(INSERT, UPDATE, DELETE) that does not support the result set. Query statements are executed in the order shown in the Query list.

To create new dataset, select [Add Dataset...] from the right mouse click popup menu on the store you created.

Enter Query dataset name in the [Add Query Dataset] dialog. New dataset product is added under the store.

UDS(User Data Store) is a Java Interface to create a dataset from various external data sources like data files (TXT, CSV, XML) or user application data (EJB, Servlet, ASP, JSP, Applet).

UDS gets data in the form of file, XML, vector, stream, or EJB object from user application and transform them into OZ data source.



Developer should write some code using UDS Java Interface to transform various user data into ResultSet object recognizable by OZ report. Java class file format of this application can be recognized by OZ report and its UDS dataset can be used as like normal SQL dataset.

OZ report can pass parameter string using user class and can get data value changing dynamically upon the parameter value.

For more details on UDS class programming, refer to "OZ API Developer's Guide".

To create a user data store, select **[Add Store...]** from the right mouse click popup menu on the USER DATA(UDS) in OZ data window.

Set UDS name with [Add User Data Store] dialog, and add the class file by using **[...]** button, then the class file(DLL) path will be set automatically.

Click **[OK]** and the new user data store appears under USER DATA(UDS) item.

Select user data store name you added under the USER DATA(UDS) in OZ data window. [Add Query dataset] dialog appears. Enter dataset name and click **[OK]**.
Select [Manual] tab of the query window and enter execution statement.

Run the query statement by selecting **[Run]** > **[Run Query]** on the menubar, or click the run query icon on the toolbar. If no error, the result data shows on the [Result] tab and result fields appears under the dataset in the data window.

When adding dataset in user data store, only the manual mode , no design view mode, is available for design mode.

Select **[Add Store...]** from the right mouse click popup menu on the USER DATA(UDS) in OZ data window. [Add User Data Store] dialog appears. Check "Use field definition manually" option to add user field to user data store.

Add dataset into your user data store named "CLASS" and select **[Set User Field...]** from the right mouse click popup menu on the new dataset.

Enter field name and field type with [Set User Field] dialog. User fields you have defined appears under dataset named "SET"

Dataset and user fields can be added without class file to be associated. When adding user fields into dataset in a user data store to which a class file was associated, only the manual mode , no design view mode, is available for design mode.

Group data store is a second data source that made from original data source (database, UDS, file/HTTP store, XML DTD store) and split into multiple datasets.

Group data store derives multiple datasets from one dataset, so it takes advantages of requesting smaller queries and data transmissions than multiple independent dataset.

For example, It would be better that create one dataset from one query and split into three datasets using group data store than creating three datasets from three queries.

Original dataset :

The dataset from which multiple datasets are derived. When split a master/detail dataset, only the top level master dataset can be the original dataset.

Group dataset :

The datasets derived from the original dataset. The group datasets can be used the same as normal dataset.

Create the original data source to get data actually.
Select **[Add Store...]** from the right mouse click popup menu on the [GROUP DATA] in OZ data window.

Enter group data store name using [Add Group Data Store] dialog and select option for "Source dataset for grouping" and click **[OK]**.

Select **[Add Dataset...]** from the right mouse click popup menu on the newly added store in OZ data window.

Enter group dataset name in [Add Group Dataset] dialog and click **[OK]**.

Enter split condition in [Add Group Dataset] dialog and click **[OK]**.

Your dataset appears under the group store.

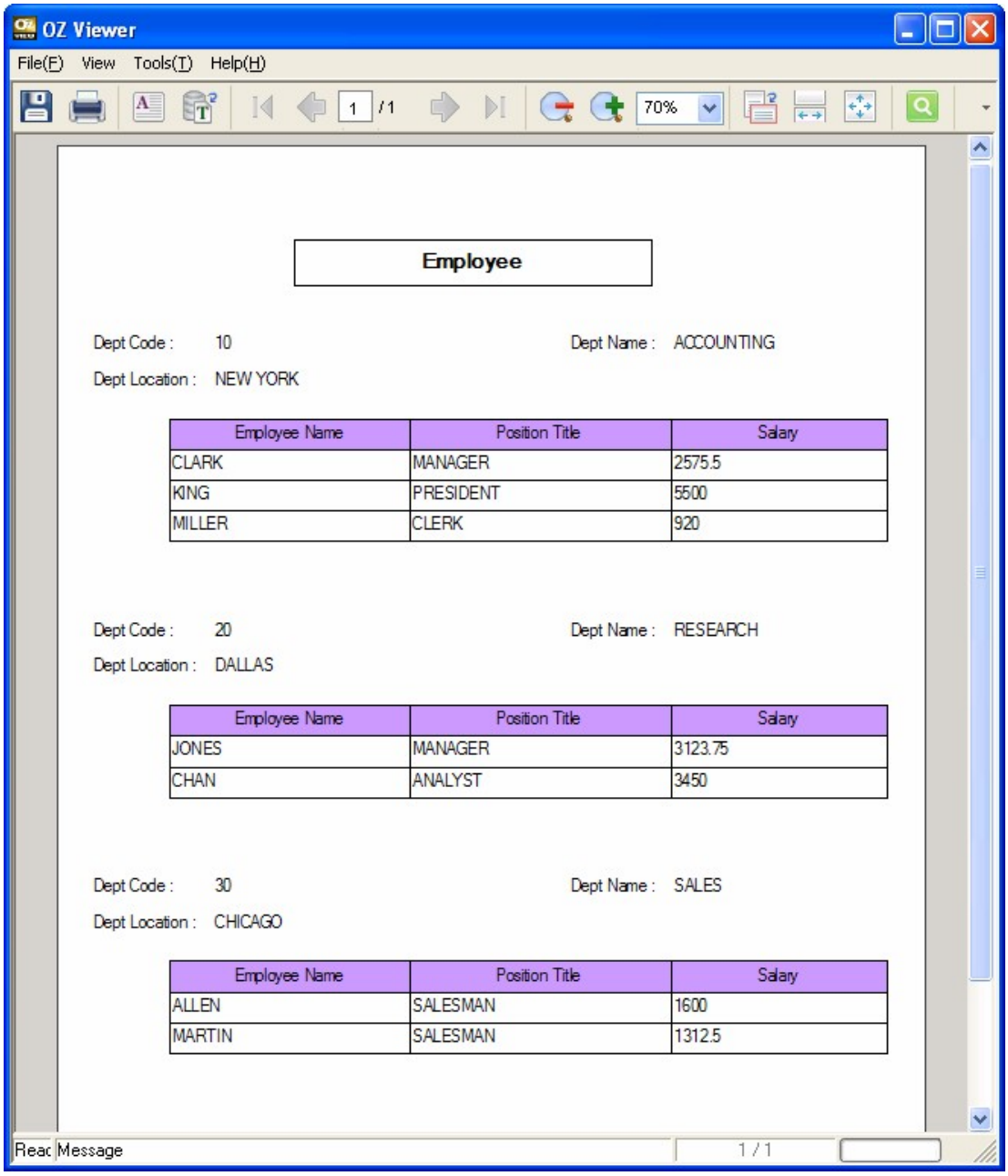
Dialog Options Description

Group dataset name :	your group dataset name
Execution statement :	split condition
Consume row :	Splits the original dataset into multiple datasets. When data splitting of a row is finished and getting next row, "True" must be set to complete group store binding. Normally, the dataset of the lowest level in a group store might be set to "True".
Add Parameter :	Configure parameters with the parameter wizard.
Select Fields :	Configure fields with [Select Fields] dialog.

The original dataset can be divided into multiple group datasets by division condition statement and vertical and horizontal division is supported.

Divide a group dataset in a group data store into multiple datasets vertically.

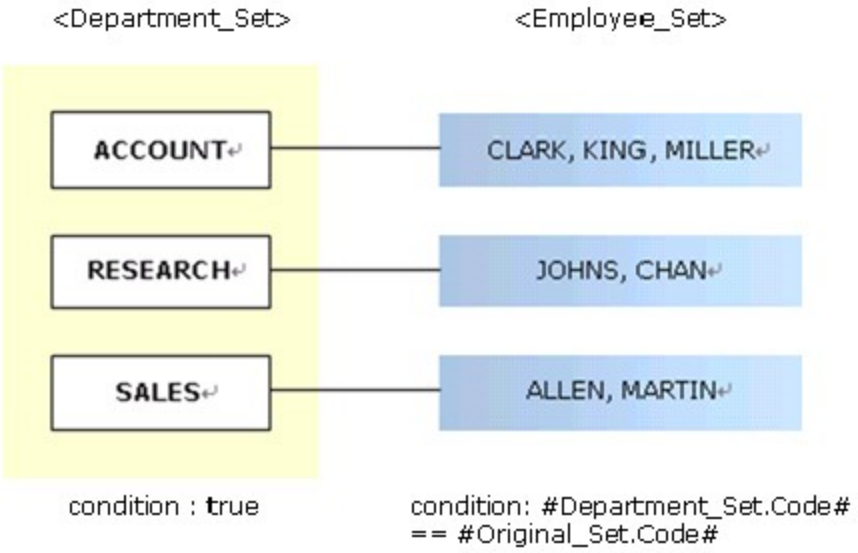
Vertical division is widely used to implement master/detail relationship. Original dataset is divided into master group dataset, detail dataset and detail group dataset.



To design above report example, first create a dataset including employee and department data.

Code	Department	Location	Employee	Position	Salary
10	ACCOUNTING	NEW YORK	CLARK	MANAGER	2572.2
10	ACCOUNTING	NEW YORK	KING	PRESIDENT	5500
10	ACCOUNTING	NEW YORK	MILLER	CLERK	920
20	RESEARCH	DALLAS	JONES	MANAGER	3123.75
20	RESEARCH	DALLAS	CHAN	ANALYST	3450
30	SALES	CHICAGO	ALLEN	SALESMAN	1600
30	SALES	CHICAGO	MARTIN	SALESMAN	1312.5

The original dataset can be vertically divided into two data group dataset, i.e., employee dataset and department dataset. Data of each dataset and division condition are as follows:



Vertical division and horizontal division can be used together.

Divide a group dataset in a group data store into multiple datasets horizontally.

Horizontal division produces multiple independent datasets from one dataset. master/detail relationship. Original dataset is divided into master group dataset, detail dataset and detail group dataset.

Lets take an example of the original dataset:

ProductCode	ProductName	UnitPrice
A	FLAT TV 19"	500,000
A	FLAT TV 21"	800,000
A	FLAT TV 25"	1,500,000
B	wide tire	100,000
B	wide tire V	180,000
C	apple flavor drink	500
C	grape flavor drink	500

To make tables by ProductCode, divide the original dataset horizontally.

<A Set> (condition) #Original_Set.ProductCode# == "A"

ProductCode	ProductName	UnitPrice
A	FLAT TV 19"	500,000
A	FLAT TV 21"	800,000
A	FLAT TV 25"	1,500,000

<B Set> (condition) #Original_Set.ProductCode# == "B"

ProductCode	ProductName	UnitPrice
B	wide tire	100,000
B	wide tire V	180,000

<C Set> (condition) #Original_Set.ProductCode# == "C"

ProductCode	ProductName	UnitPrice
C	apple flavor drink	500
C	grape flavor drink	500

Vertical division and horizontal division can be used together.

File store enables you to use text file as data source. CSV file and standard XML file with no DTD or XSD can be used as data sources.

XML dataset, CSV dataset, dynamic XML dataset, dynamic CSV dataset are can be created in the file store.

To create a XML dataset, select **[Add XML Dataset...]** from the right mouse click popup menu on the FILE STORE in OZ data window.

Set options in [Add XML Dataset - FILESTORE] dialog. Click **[OK]** on [Add XML Dataset - FILESTORE] dialog and XML dataset information appears under file store in data window.

Dialog Options Description

XML dataset name :	dataset name
XML file location :	XML file path on the server. It can be set with the parameter wizard by clicking **[Parameter]** button and also be selected by **[Browse]** button.
Temp XML file location in design :	A temporary design time XML file path on the local system. It can be set with the parameter wizard by clicking **[Parameter]** button and also can be selected by **[Browse]** button.

Dataset node name :	The node name representing the dataset in the XML file. Default name is DATASET.
Recordset node name :	The node name representing the record set in the XML file. Default name is RECORD.
Alias :	Check if use dataset alias defined in datasource.properties file on the server and enter the alias name. **Alias filename :** configuration file for dataset alias.
View Data :	Preview data of the XML file.
Save Dataset Information :	Save dataset information (field name, field type) as XML file. This button is activated only when modifying the dataset and uses UTF-8 data encoding.

To create a CSV dataset, select **[Add CSV Dataset...]** from the right mouse click popup menu on the FILE STORE in OZ data window.

Set options [Add CSV Dataset - FILESTORE] dialog. Click **[OK]** on [Add CSV Dataset - FILESTORE] dialog and CSV dataset information appears under file store in data window.

Dialog Options Description

CSV dataset name :	dataset name
CSV file location :	CSV file path on the server. It can be set with the parameter wizard by clicking **[Parameter]** button and also be selected by **[Browse]** button.
Temp CSV file location in design :	A temporary design time CSV file path on the local system. It can be set with the parameter

	wizard by clicking **[Parameter]** button and also be selected by **[Browse]** button.
Field delimiter :	Set the field delimiter. Select one of TAB, SPACE, SEMICOLON, COLON, COMMA, or enter your delimiter. You can also set the field delimiter with the parameter wizard by clicking **[Parameter]** button.
Include field type information :	Check if CSV file includes field type. If not check, all field types are regarded as "VARCHAR".
Line delimiter :	Set the line delimiter. You can enter your delimiter and the default is LINEFEED. You can also set the line delimiter with the parameter wizard by clicking **[Parameter]** button.
Null mark :	A string to represent the NULL value.
Character set :	Character set of a file. The default is default locale of the system.
Error indication code :	Set the code to notice a data error. If the code value appears at the first line of the file, an error message box will be displayed with no report preview. For example, if a CSV file have "ERR" at the first line and you set the error indication code to "ERR", "ERR" won't be recognized as data and a message box shows saying the file is not valid CVS file.
Enclosed with Double Quotes :	Check if the data enclosed in double quotes is treated as one data.
Alias :	Check if use dataset alias defined in datasource.properties file on the server and enter the alias name. **Alias filename :** configuration file for dataset alias
View Data :	Preview data of the CSV file.
Save Dataset Information :	Save dataset information (field name, field type) as CSV file. This button is activated only when

modifying the dataset. Check "Include field type information" option to save field type information with data.

To create a dynamic XML dataset, select **[Add Dynamic Dataset...]** from the right mouse click popup menu on the FILE STORE in OZ data window.

Check "XML dataset" in [Add Dynamic Dataset - File Store] dialog.

Set options [Add Dynamic Dataset - XML] dialog. Click **[OK]** on [Add Dynamic Dataset - XML] dialog and dynamic XML dataset information appears under file store in data window.

Dialog Options Description

XML dataset name :	dataset name
Starting row number :	The starting number of ROW INDEX for dynamic dataset. The value must be in range of 0~2147483647 and the default is zero.
XML file location :	XML file path on the server. It can be set with the parameter wizard by clicking **[Parameter]** button and also be selected by **[Browse]** button.
Temp XML file location in design :	A temporary design time XML file path on the local system. It can be set with the parameter wizard by clicking **[Parameter]** button and also be selected by **[Browse]** button.
Dataset node name :	The node name representing the dataset in the XML file. Default name is DATASET.
Recordset node name :	The node name representing the record set in the XML file. Default name is RECORD.

Alias :	Check if use dataset alias defined in datasource.properties file on the server and enter the alias name. **Alias filename :** configuration file for dataset alias
View Data :	Preview data of the XML file.
Save Dataset Information :	Save dataset information (field name, field type) as XML file. This button is activated only when modifying the dataset and uses UTF-8 data encoding.

To create a dynamic CSV dataset, select **[Add Dynamic Dataset...]** from the right mouse click popup menu on the FILE STORE in OZ data window.

Check "CSV dataset" in [Add Dynamic Dataset - File Store] dialog.

Set options [Add Dynamic Dataset - CSV] dialog. Click **[OK]** on [Add Dynamic Dataset - CSV] dialog and dynamic CSV dataset information appears under file store in data window.

Dialog Options Description

CSV dataset name :	dataset name
Starting row number :	The starting number of ROW INDEX for dynamic dataset. The value must be in range of 0~2147483647 and the default is zero.
CSV file location :	CSV file path on the server. It can be set with the parameter wizard by clicking **[Parameter]** button and also be selected by **[Browse]** button.

Temp CSV file location in design : A temporary design time CSV file path on the local system. It can be set with the parameter wizard by clicking **[Parameter]** button and also be selected by **[Browse]** button.

Field delimiter : Set the field delimiter. Select one of TAB, SPACE, SEMICOLON, COLON, COMMA, or enter your delimiter.
You can also set the field delimiter with the parameter wizard by clicking **[Parameter]** button.

Include field type information : Check if CSV file includes field type. If not check, all field types are regarded as "VARCHAR".

Line delimiter : Set the line delimiter. You can enter your delimiter and the default is LINEFEED.
You can also set the line delimiter with the parameter wizard by clicking **[Parameter]** button.

Null mark : A string to represent the NULL value.

Character set : Character set of a file. The default is default locale of the system.

Error indication code : Set the code to notice a data error. If the code value appears at the first line of the file, an error message box will be displayed with no report preview. For example, if a CSV file have "ERR" at the first line and you set the error indication code to "ERR", "ERR" won't be recognized as data and a message box shows saying the file is not valid CVS file.

Enclosed with Double Quotes : Check if the data enclosed in double quotes is treated as one data.

Alias : Check if use dataset alias defined in datasource.properties file on the server and enter the alias name.
Alias filename : configuration file for dataset alias

View Data : Preview data of the CSV file.

Save Dataset Information : Save dataset information (field name, field type) as CSV file. This button is activated only when modifying the dataset. Check "Include field type information" option to save field type information with data.

Describes specifications for XML and CSV file to be used to create dataset in the file store.

Extensible Markup Language (XML), defined by The World Wide Web Consortium(W3C), is the standard format for data exchange between applications on the Web while HTML is a language for User Interface development.

OZ provides file store and XML store to use XML as data source. File store is used for simple XML with no DTD or XSD and XML store is used for XML with DTD or XSD.

XML file specifications for file store are:

- Supports standard XML file specification without DTD.
- Nodes that contain dataset and record set are required.
- Nodes that contain field information should follow the format below and can be omitted.

Following is an example of XML file that contains dataset node named "DATASET", record set node named "RECORD" and field information node.

```
<?xml version="1.0" encoding="KSC5601"?>
<XMLSET>
<DATASET>
<FIELDINFO>
<FIELD NAME="ID" TYPE="INT"/>
<FIELD NAME="NAME" TYPE="CHAR"/>
```

```
<FIELD NAME="ADDRESS" TYPE="VARCHAR"/>
</FIELDINFO>
<RECORD>
<ID>1</ID>
<NAME>Park Chan Ho</NAME>
<ADDRESS>Texas</ADDRESS>
</RECORD>
<RECORD>
<ID>2</ID>
<NAME>Shin Ki Sung</NAME>
<ADDRESS>Seoul</ADDRESS>
</RECORD>
<RECORD>
<ID>3</ID>
<NAME>Kim Byung Hyun</NAME>
<ADDRESS>Arizona</ADDRESS>
</RECORD>
</DATASET>
</XMLSET>
```

CSV (Comma separated value) file is widely used for data exchange between databases. OZ file store supports not only the comma but also the user defined delimiter.

CSV file specifications for file store are:

- Supports standard XML file specification without DTD.
- First row contains field names, second row contains field types and data record begins from 3rd row.
- Fields are separated by the field delimiter and rows are separated by new line.
- The second row that contains field types can be omitted.

Following is an example of CSV file using comma(",") as the field delimiter.

```
no,jumin_id,name,address
INT,VARCHAR,VARCHAR,VARCHAR
1,7411231932347,kim,seoul
2,6305212983717,lee,pusan
3,7006031837261,park,daegu
4,7271722848281,song,daejeon
```

HTTP store enables you to use HTTP output as data source. CSV file and standard XML file with no DTD or XSD can be used as data sources.

XML dataset, CSV dataset, dynamic XML dataset, dynamic CSV dataset are can be created in the HTTP store.

To create a XML dataset, select **[Add XML Dataset...]** from the right mouse click popup menu on the HTTP STORE in OZ data window.

Set options in [Add XML Dataset - HTTPSTORE] dialog. Click **[OK]** on [Add XML Dataset - HTTPSTORE] dialog and XML dataset information appears under HTTP store in data window.

Dialog Options Description

XML dataset name :	Dataset name
User application URL :	HTTP address of user application or data file. User application URL can also be set using the parameter wizard by clicking **[Parameter]** button.
Temp XML file location in design :	A temporary design time XML file path on the local system. It can be set with the parameter

	wizard by clicking **[Parameter]** button and also be selected by **[Browse]** button.
Dataset node name :	The node name representing the dataset in the XML file. Default name is DATASET.
Recordset node name :	The node name representing the record set in the XML file. Default name is RECORD.
Server Log file name :	The file on the server to log data received when ODI is binding on the server. Enter only the file name without path.
Retry count :	Retry count to reconnect when an error occurs with HTTP store dataset. If not set or invalid value, the default is 5.
Set URL encoding :	Check to encode Server URL. Check if URL parameter includes foreign language character like Korean when requesting data.
Submit by POST method :	Check to use POST method to send URL parameter and uncheck for GET method.
Alias :	Check if use dataset alias defined in datasource.properties file on the server and enter the alias name. **Alias filename :** configuration file for dataset alias.
View Data :	Previews data of the XML file. If the server log file name was set, all data rows are fetched before viewing data regardless of the maximum row number and so, performance may possibly be slower.
Save Dataset Information :	Save dataset information (field name, field type) as XML file. This button is activated only when modifying the dataset and uses UTF-8 data encoding.
Save Original Data :	Save the original dataset as a file. **[Save Original Data]** button will be activated only when modifying the dataset.

To create a CSV dataset, select **[Add CSV Dataset...]** from the right mouse click popup menu on the HTTP STORE in OZ data window.

Set options in [Add CSV Dataset - HTTPSTORE] dialog. Click **[OK]** on [Add CSV Dataset - HTTPSTORE] dialog and CSV dataset information appears under HTTP store in data window.

Dialog Options Description

CSV dataset name :	dataset name
User application URL :	HTTP address of user application or data file. User application URL can also be set using the parameter wizard by clicking **[Parameter]** button.
Temp CSV file location in design :	A temporary design time CSV file path on the local system. It can be set with the parameter wizard by clicking **[Parameter]** button and also be selected by **[Browse]** button.
Field delimiter :	Set the field delimiter. Select one of TAB, SPACE, SEMICOLON, COLON, COMMA, or enter your delimiter. You can also set the field delimiter with the parameter wizard by clicking **[Parameter]** button.
Include field type information :	Check if CSV file includes field type. If not check, all field types are regarded as "VARCHAR".
Line delimiter :	Set the line delimiter. You can enter your delimiter and the default is LINEFEED. You can also set the line delimiter with the parameter wizard by clicking **[Parameter]** button.
Null mark :	A string to represent the NULL value.

Character set : Character set of a file. The default is default locale of the system.

Error indication code : Set the code to notice a data error. If the code value appears at the first line of the file, an error message box will be displayed with no report preview. For example, if a CSV file have "ERR" at the first line and you set the error indication code to "ERR", "ERR" won't be recognized as data and a message box shows saying the file is not valid CVS file.

Enclosed with Double Quotes : Check if the data enclosed in double quotes is treated as one data.

Server Log file name : The file on the server to log data received when ODI is binding on the server. Enter only the file name without path.

Retry count : Retry count to reconnect when an error occurs with HTTP store dataset. If not set or invalid value, the default is 5.

Set URL encoding : Check to encode Server URL. Check if URL parameter includes foreign language character like Korean when requesting data.

Submit by POST method : Check to use POST method to send URL parameter and uncheck for GET method.

Alias : Check if use dataset alias defined in datasource.properties file on the server and enter the alias name.

Alias filename : configuration file for dataset alias

View Data : Previews data of the CSV file. If the server log file name was set, all data rows are fetched before viewing data regardless of the maximum row number and so, performance may possibly be slower.

Save Dataset Information : Save dataset information (field name, field type) as CSV file. This button is activated only when modifying the dataset.

Save Original Data :	Save the original dataset as a file. **[Save Original Data]** button will be activated only when modifying the dataset.

To create a dynamic XML dataset, select **[Add Dynamic Dataset...]** from the right mouse click popup menu on the HTTP STORE in OZ data window.

Check "XML dataset" in [Add Dynamic Dataset - HTTP Store] dialog.

Set options in [Add Dynamic Dataset - XML] dialog. Click **[OK]** on [Add Dynamic Dataset - XML] dialog and dynamic XML dataset information appears under HTTP store in data window.

Dialog Options Description

XML dataset name :	Dataset name
Starting row number :	The starting number of ROW INDEX for dynamic dataset. The value must be in range of 0~2147483647 and the default is zero.
User application URL :	HTTP address of user application or data file. User application URL can also be set using the parameter wizard by clicking **[Parameter]** button.
Temp XML file location in design :	A temporary design time XML file path on the local system. It can be set with the parameter wizard by clicking **[Parameter]** button and also be selected by **[Browse]** button.
Dataset node name :	The node name representing the dataset in the XML file. Default name is DATASET.
Recordset node name :	The node name representing the record set in the XML file. Default name is RECORD.

Server Log file name :	The file on the server to log data received when ODI is binding on the server. Enter only the file name without path.
Retry count :	Retry count to reconnect when an error occurs with HTTP store dataset. If not set or invalid value, the default is 5.
Set URL encoding :	Check to encode Server URL. Check if URL parameter includes foreign language character like Korean when requesting data.
Submit by POST method :	Check to use POST method to send URL parameter and uncheck for GET method.
Alias :	Check if use dataset alias defined in datasource.properties file on the server and enter the alias name. **Alias filename :** configuration file for dataset alias.
View Data :	Previews data of the XML file. If the server log file name was set, all data rows are fetched before viewing data regardless of the maximum row number and so, performance may possibly be slower.
Save Dataset Information :	Save dataset information (field name, field type) as XML file. This button is activated only when modifying the dataset and uses UTF-8 data encoding.
Save Original Data :	Save the original dataset as a file. **[Save Original Data]** button will be activated only when modifying the dataset.

To create a dynamic CSV dataset, select **[Add Dynamic Dataset...]** from the right mouse click popup menu on the HTTP STORE in OZ data window.

Check "CSV dataset" in [Add Dynamic Dataset - HTTP Store] dialog.

Set options in [Add Dynamic Dataset - CSV] dialog. Click **[OK]** on [Add Dynamic Dataset - CSV] dialog and dynamic CSV dataset information appears under HTTP store in data window.

Dialog Options Description

CSV dataset name :	Dataset name
Starting row number :	The starting number of ROW INDEX for dynamic dataset. The value must be in range of 0~2147483647 and the default is zero.
User application URL :	HTTP address of user application or data file. User application URL can also be set using the parameter wizard by clicking **[Parameter]** button.
Temp CSV file location in design :	A temporary design time CSV file path on the local system. It can be set with the parameter wizard by clicking **[Parameter]** button and also be selected by **[Browse]** button.
Field delimiter :	Set the field delimiter. Select one of TAB, SPACE, SEMICOLON, COLON, COMMA, or enter your delimiter. You can also set the field delimiter with the parameter wizard by clicking **[Parameter]** button.
Include field type information :	Check if CSV file includes field type. If not check, all field types are regarded as "VARCHAR".
Line delimiter :	Set the line delimiter. You can enter your delimiter and the default is LINEFEED. You can also set the line delimiter with the parameter wizard by clicking **[Parameter]** button.
Null mark :	A string to represent the NULL value.
Character set :	Character set of a file. The default is default locale of the system.

Error indication code : Set the code to notice a data error. If the code value appears at the first line of the file, an error message box will be displayed with no report preview. For example, if a CSV file have "ERR" at the first line and you set the error indication code to "ERR", "ERR" won't be recognized as data and a message box shows saying the file is not valid CVS file.

Enclosed with Double Quotes : Check if the data enclosed in double quotes is treated as one data.

Server Log file name : The file on the server to log data received when ODI is binding on the server. Enter only the file name without path.

Retry count : Retry count to reconnect when an error occurs with HTTP store dataset. If not set or invalid value, the default is 5.

Set URL encoding : Check to encode Server URL. Check if URL parameter includes foreign language character like Korean when requesting data.

Submit by POST method : Check to use POST method to send URL parameter and uncheck for GET method.

Alias : Check if use dataset alias defined in datasource.properties file on the server and enter the alias name.

Alias filename : configuration file for dataset alias.

View Data : Previews data of the CSV file. If the server log file name was set, all data rows are fetched before viewing data regardless of the maximum row number and so, performance may possibly be slower.

Save Dataset Information : Save dataset information (field name, field type) as CSV file. This button is activated only when modifying the dataset and uses UTF-8 data encoding.

Save Original Data : Save the original dataset as a file. **[Save Original Data]** button will be activated only when modifying the dataset.

SOAP(Simple Object Access Protocol) is a lightweight protocol intended for exchanging structured information in a distributed environment. SOAP uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols.

OZ uses WSDL(Web Service Description Language, an XML format for describing web services) to build web service applications

Select **[Add Store...]** from the right mouse click popup menu on the [SOAP STORE] in OZ data window.

Set SOAP Store Name and WSDL Location using [Add SOAP Store] dialog. Click **[Get Services]** button and configure Namespace and Services. Click **[OK]** on [Add SOAP Store] dialog and XML dataset information appears under SOAP store in data window.

Dialog Options Description

SOAP Store Name :	SOAP store name
WSDL Location :	WSDL document location
Alias :	Check if use dataset alias defined in datasource.properties file on the server and enter the alias name. **Alias filename :** configuration file for dataset alias.
Get Services :	Show name space list and service list. The button is activated only when setting the **WSDL Location** option. After connecting to WSDL location, Name space and service list will be shown.

Namespace : Show Prefix and NameSpace URI defined in a WSDL file. Set the Prefix to use.

Services : Service list.

Service : Select a service from combo box list.

Port : Select a port from binding list of the service.

Operation : Select an operation from binding list of the port.

SOAP Action : SOAP action of the operation.

Request : Enter name, type, XPath, OZParam for the operation in the format of "#OZParam.parameter_name#".

Preview Request : Show SOAP message in XML.

SOAP Header : Open a dialog to enter SOAP Header.

Auto Mapping : Automatically set "OZParam" value of requested option in the form of "#OZParam.<operation name>_<name>#".

SOAP store feature requires SOAP Toolkit 3.0 library. Register MSSOAP30.dll and WHSC30.dll to the registry by executing bin/soap/register_soap.bat under OZ Query Designer installation directory.

Manual design mode cannot be changed to design view mode.
If automatically set a user parameter field, "DateTime" type of XML schema is considered as "VARCHAR".
To define logical types in a WSDL message, part and type should get 1:1 mapping, but for n:1 WSDL, 1st part and type would get 1:1 mapping.

Describes how to add a dataset in SOAP store for design view mode and manual design mode.

To create a dataset, select **[Add Dataset...]** from the right mouse click popup menu on the SOAPStore_ in OZ data window. Add dataset by entering dataset name in [Add Query Dataset] dialog. The node structure of SOAP store will appears in the node tree window of the query window.

Each node has an icon in front that show the characteristics of the node, which makes it easier to identify. The following table contains description of the icons.

	Attribute
	An element of 1 minimum occurrence and 1 maximum occurrence (default)
	An element of 0 minimum occurrence and unlimited maximum occurrence
	An element of 0 minimum occurrence and 1 maximum occurrence
	An element of 1 minimum occurrence and unlimited maximum occurrence
	An element expanded from the tree
	An element not selectable as a field
	An element defined as the base XPath

Click **[Set Base XPath]** from the right mouse click popup menu on the node to be designated as the base XPath in the node tree window.

The icon of the base XPath node is changed to checked icon and the node name is highlighted and the base XPath is shown on the textbox of bottom line.

To change the base XPath, edit it in the textbox, or select **[Set base XPath]** from the right mouse click popup menu on the node which is supposed to be the new base XPath.

Double click a node in the node tree window and add dataset fields. [Selected Field] window shows the added fields.

With the selected field window, you can change the field name, change the order by [Top], [Up], [Down], [Bottom] icon, and delete a field by [Delete] icon.

Click the run icon after adding all fields and all new fields appears.

To create a dataset, select **[Add Dataset...]** from the right mouse click popup menu on the SOAPStore_ in OZ data window. Add dataset by entering dataset name in [Add Query Dataset] dialog.

Change the design mode of the query window to the manual mode by selecting **[Query] > [Design Mode] > [Manual]**. The node structure of SOAP store should appear in the node tree window of the query window.

Each node has an icon in front that show the characteristics of the node, which makes it easier to identify. The following table contains description of the icons.

	Attribute
	An element of 1 minimum occurrence and 1 maximum occurrence (default)
	An element of 0 minimum occurrence and unlimited maximum occurrence
	An element of 0 minimum occurrence and 1 maximum occurrence
	An element of 1 minimum occurrence and unlimited maximum occurrence
	An element expanded from the tree
	An element not selectable as a field
	An element defined as the base XPath

To set the base XPath, enter XPath string in the textbox, or select **[Copy]** from the right mouse click popup menu on the node which is supposed to be the new base XPath and paste it to the textbox.

Double click a node in the node tree window and add dataset fields. [Selected Field] window shows the added fields.

With the selected field window, you can change the field name, change the order by [Top], [Up], [Down], [Bottom] icon, and delete a field by [Delete] icon.

Click the run icon after adding all fields and all new fields appears.

ADO DataSet web service store is used to access ADO DataSet object in .NET web service.

To create a store, select **[Add Store...]** from the right mouse click popup menu on the **[ADO DataSet WebService]** in OZ data window.

Set Store Name and WSDL Location in [Add ADO DataSet Webservice] dialog. ADO DataSet Webservice store appears under ADO DataSet Webservice in the data window.

Dialog Options Description

Store Name : ADO DataSet Webservice store name

WSDL Location : WSDL document Location can be set with the parameter wizard by clicking **[Parameter]** button.

To create a dataset, select **[Add Dataset...]** from the right mouse click popup menu on the store you added.

Enter ADO Dataset Name in ADO Dataset dialog. The query window is changed to configure ADO DataSet web service dataset.

Click the run icon after dataset configuration. [ADO Dataset Information] dialog appears and set SET_NAME. Datasets and fields appear under ADOstore.

Dialog Options Description

WebMethod :	Web Methods defined in WSDL are listed in the combobox.
OZParam :	Show and set parameter name defined in WSDL.
Param AutoMapping :	Set parameter value to in the format of "#OZParam.parameter_name#".

XML (Extensible Markup Language), defined by W3C (The World Wide Web Consortium), is the universal format for exchanging data between applications on the Web. XML does not replace HTML which is HyperText Markup Language for webpage User Interface design.

DTD (Document Type Definition) defines the structure of an XML document and XSD (XML schema definition) describes the structure of an XML document. DTD and XSD define tags and inclusion relationship between them, the order of tags, the number of tags cab be included, and tag attributes and its values.

XML specification verifies document by XML structure information such as DTD/XSD at XML analysis step, so the application using XML data can be free from data validation. The application using XML data takes advantage of easy data exchange with other applications.

OZ Report provides XML store for XML based data source. XML store supports XML document with DTD/XSD while FILE/HTTP store supports XML document without DTD/XSD. XML store can have two forms, i.e., XML DTD store and XML XSD store.

To create a XML store, select **[Add XML Data Store...]** from the right mouse click popup menu on the [XML] in OZ data window.

Set options in [Add XML Store] dialog and click **[OK]**. Database information appears under [XML] in data window.

Dialog Options Description

XML store name : XML store name

XML address (File path or URL) :	XML file path or application path on the server. It can be set with the parameter wizard by clicking **[Parameter]** button and can also be selected by **[Browse]** button.
Temp XML file location in design :	A temporary design time XML file path on the local system. It can be set with the parameter wizard by clicking **[Parameter]** button and also can be selected by **[Browse]** button.
XML schema file path in design time (DTD or XSD) :	A design time DTD/XSD XML file path on the local system. It can be set with the parameter wizard by clicking **[Parameter]** button and also can be selected by **[Browse]** button.
XML parser(Java) :	Sets the XML parser. This function is only available in Java server. **xerces :** Set xerces. **dom4j :** Set dom4j.
Server Log file name :	The file on the server to log XML data when parsing error occurs. Data logging works only when the XML address is http:// or https:// type and XMLSTORE_LOG_FILE option value defined in log.properties file on the server is MANUAL or AUTO. If check **Alias** option, **server Log file name** option will be inactivated and the server file log name will be alias.FILENAMETOLOG option value in datasource.properties file, not the value set in [Add XML Store] dialog. The actual log file name consist of the server log file name, date, time and serial number. The log file path is XMLSTORE_LOG_FILE.LOCATION option value defined in log.properties file.
Set URL encoding :	Check to encode Server URL. Check if URL parameter includes foreign language character

like Korean when requesting data.

Submit by POST method : Check to use POST method to send URL parameter and uncheck for GET method.

Alias : Check if use dataset alias defined in datasource.properties file on the server and enter the alias name.
Alias filename : configuration file for dataset alias.

Describes how to add a dataset in XML store for design view mode and manual design mode.

Select **[Add Dataset...]** from the right mouse click popup menu on the XMLEXStore_ in OZ data window. Add dataset by entering dataset name and click **[OK]** in the dialog. The node structure of XML store will appears in the node tree window of the query window.

Each node has an icon in front that show the characteristics of the node, which makes it easier to identify. The following table contains description of the icons.

	Attribute
	An element of 1 minimum occurrence and 1 maximum occurrence (default)
	An element of 0 minimum occurrence and unlimited maximum occurrence
	An element of 0 minimum occurrence and 1 maximum occurrence
	An element of 1 minimum occurrence and unlimited maximum occurrence
	An element expanded from the tree

	An element not selectable as a field
	An element defined as the base XPath

Click **[Set Base XPath]** from the right mouse click popup menu on the node to be designated as the base XPath in the node tree window.

The icon of the base XPath node is changed to checked icon and the node name is highlighted and the base XPath is shown on the textbox of bottom line.

To change the base XPath, edit it in the textbox, or select **[Set base XPath]** from the right mouse click popup menu on the node which is supposed to be the new base XPath.

Double click a node in the node tree window and add dataset fields. [Selected Field] window shows the added fields.

With the selected field window, you can change the field name, change the order by [Top], [Up], [Down], [Bottom] icon, and delete a field by [Delete] icon.

Click the run icon after adding all fields and all new fields appears.

Select **[Add Dataset...]** from the right mouse click popup menu on the XMLEXStore_ in OZ data window. Add dataset by entering dataset name and click [OK] in the dialog.

Change the design mode of the query window to the manual mode by selecting **[Query] > [Design Mode] > [Manual]** in the menubar. The node structure of XML store should appear in the node tree window of the query window.

Each node has an icon in front that show the characteristics of the node, which makes it easier to distinguish. The following table contains description of the icons.

	Attribute
	An element of 1 minimum occurrence and 1 maximum occurrence (default)

	An element of 0 minimum occurrence and unlimited maximum occurrence
	An element of 0 minimum occurrence and 1 maximum occurrence
	An element of 1 minimum occurrence and unlimited maximum occurrence
	An element expanded from the tree
	An element not selectable as a field
	An element defined as the base XPath

To set the base XPath, enter XPath string in the textbox, or select **[Copy]** from the right mouse click popup menu on the node which is supposed to be the new base XPath and paste it to the textbox.

Double click a node in the node tree window and add dataset fields. [Selected Field] window shows the added fields.

With the selected field window, you can change the field name, change the order by [Top], [Up], [Down], [Bottom] icon, and delete a field by [Delete] icon.

Click the run icon after adding all fields and all new fields appears.

SAP provides an integrated business application with database and SAP R/3 supports a work flow feature distributing documents to users. SAP basically runs on client/server environments and most of operating systems.

Generally, to create a report using SAP database, the user have to create a data extracting program and install the SAP client application to call the data extracting program. OZ SAP store enables the user can easily create a dataset without those extra programs.

To extract data from SAP store, use RFC (Remote Function Call) interface to SAP R/3.

To create a SAP store, select **[Add SAP Store...]** from the right mouse click popup menu on the **[SAP]** in OZ data window.

Set options in [SAP Connection Information] dialog and click **[OK]**. The SAP store created appears under [SAP] in the data window.

Dialog Options Description

SAP Store Name :	SAP store name
Client :	Client number to login
User ID :	User id to login
User Password :	Password to login. A password can also be set to the user defined parameter using parameter wizard.
Language :	language to use
Host Name(or IP Address) :	SAP server IP Address

NO. System :	System number of a SAP server.
SAP alias :	Check if use an SAP store alias defined in sap.properties file on the server. **Alias information filename :** configuration file for SAP alias.

To create a dataset, select **[Add Function...]** from the right mouse click popup menu on the store you added.

Enter search condition to textbox with "*" and/or "?" in [SAP RFC Set] dialog and click **[Search]** or click **[Search]** after checking "All Function List" option.
Select and double click your RFC from RFC list and click **[OK]**.

Set the parameter value for the selected RFC in [Set SAP RFC Parameter] dialog.

Set dataset information in [Set SAP RFC Dataset Name] dialog and click **[OK]**. The dataset appears under [SAPStore_] in the data window.

CLEAR QUEST store enables you to use a database created by IBM Rational Clear Quest, a configuration management tool, as a data source.

Restrictions on using CLEAR QUEST store :

- Adding query mode dataset supported by CLEAR QUEST is not currently supported and filter definition statement of dynamic mode dataset is not shown.
- If the user who logged in CLEAR QUEST system does not have the permission to see SQL, the manual design mode is not available.
- CLEAR QUEST store requires IBM Rational Clear Quest installation on user's local computer.
- CLEAR QUEST store does not shows available table list in the data window.
- Cannot change from the manual design mode to design view mode.

CLEAR QUEST store can be added to the Design view mode and Manual mode.

Design view mode

To create a CLEAR QUEST store, select **[Add Store...]** from the right mouse click popup menu on the [CLEAR QUEST] in OZ data window.

Set options such as ClearQuest Store Name, User name, Password and Database in the [Add ClearQuest Store] dialog and click **[OK]**. The store created appears under [CLEAR QUEST] in the data window.

Manual mode

Select **[Add Store...]** from the right mouse click popup menu on the [CLEAR QUEST] in OZ data window.

Enter CLEAR QUEST store name in [Add ClearQuest Store] dialog and click **[Get connection information]** on the "Schema Repository List" option. The list of schema repository added in CLEAR QUEST will be shown.

Double click the schema to connect in the list of schema repository and gets database list. Set user name, password and database and click **[OK]**. The store created appears under [CLEAR QUEST] in the data window.

CLEAR QUEST dataset can be added to the Design view mode and Manual mode.

Design view mode - record type

Creates a CLEAR QUEST dataset in association with CLEAR QUEST record type.

Select **[Add Dataset...]** from the right mouse click popup menu on the store you created under [CLEAR QUEST] in OZ data window. Select [Record Type] tab in the [Add ClearQuest Dataset] dialog.

Enter dataset name in [Add ClearQuest Dataset] dialog, select the record type to connect, and click **[OK]**. Dataset will be added and the user can configure dataset in [Query] tab.

Select fields from the [Available fields] list and add them to [Selected fields] list using ([Add], [Delete]) buttons and set the order of fields using ([Up], [Down]) buttons.

After adding from [Available fields] list to [Selected Filter] list, select field using ([Add], [Delete]) buttons and change the operator of selected field using operator buttons ([OR], [AND]).

To set values in the filter condition, enter the value by yourself or use the parameter wizard.

After finishing with options, run the query by **[Run Query]** button on the toolbar.

Design view mode - workspace

Creates a CLEAR QUEST dataset in association with CLEAR QUEST workspace.

Select **[Add Dataset...]** from the right mouse click popup menu on the store you created under [CLEAR QUEST] in OZ data window. Select [Workspace] tab in the [Add ClearQuest Dataset] dialog.

Enter dataset name in [Add ClearQuest Dataset] dialog, select the workspace to connect, and click **[OK]**. Dataset will be added and the user can configure dataset in [Query] tab.

After finishing with options, run the query by **[Run Query]** button on the toolbar.

Manual mode

Select **[Add Dataset...]** from the right mouse click popup menu on the store you created under [CLEAR QUEST] in OZ data window.

Enter dataset name in the [Add ClearQuest Dataset] dialog.

Enter SQL in the [Query] tab.

Run the query by **[Run Query]** button on the toolbar.

This chapter describes how to design datasets in the manual mode and design view mode of the query window.

Select a dataset from the data window. Select **[Query] > [Design Mode] > [Manual]** from the menubar or click the **[manual icon]** on the toolbar. The query window will be switched to the manual mode. Enter your query statement in the query window.

Writing query

Switch to the manual mode and select [Query] tab and enter SQL statement.

Executing query

To execute SQL statement, select **[Run] > [Run Query]** from the menubar or click the **[Run Query]** icon on the toolbar or push the <F5> key. If no SQL error, the result will be shown in the [Result] tab.

Dataset fields

If no SQL error, the result dataset fields will be shown.

If SQL statement have comments and that includes user defined parameter, user defined parameter delimiter, and field value of master set, then the SQL will possibly not work.

Design view mode helps the user who is not familiar with SQL to design SQL easily using by the query wizard.

Select a dataset from the data window. Select **[Query] > [Design Mode] > [Design View]** from the menubar or click the **[manual icon]** on the toolbar. The query window will be switched to the design view mode.

[Design] tab of the design view mode consist of the field area on the top, table area on the middle, and query area on the bottom.

To run the Query Wizard, click the right mouse button on the [Design] tab and then click the [Query Wizard] menu.

Select [Design] tab. Drag & drop your tables from the table list of the data window to the table area of the query window.

The selected tables will be listed in SQL FROM clause after adding fields.

The join conditions can be generated automatically by OZ Query Designer or you can configure the join conditions by yourself.

Creating join conditions automatically

When the "Detect join condition automatically" option in [Options] dialog ([File] > [Options] > [database] > [General]) was checked, and if the added tables can have join condition, then the join conditions are created automatically.

For the case of a database, like Oracle or Ms-SQL, that is allowed to get the Primary and Foreign Key combination via JDBC functions, if fields of two table are connected by Primary or Foreign Key, the join condition will be created automatically.
Although a database is not allowed to get the Primary and Foreign Key combination via JDBC functions, if fields of two table have the same name and same type, the join condition will also be created automatically.

Creating join conditions manually

To create join conditions manually, connect from the source table field to target table field by dragging. If select and double click the join connection line, or select **[Join Type]** from the right mouse click popup menu on the join connection line, [Join Type] dialog will appears.

You can change the type of join using the Join Type dialog. Join Operator can be set to "=", "<", "<=", ">", ">=", "<>". Join Type can be set to "Inner Join", "Left Outer Join", "Right Outer Join", "Full Outer Join".

Deleting join conditions

To delete join conditions, select the join connection line and push <delete> key, or select **[Delete Join]** from the right mouse click popup menu on the join connection line.

[Option] dialog appears by selecting **[File] > [Option]** in the menubar. And the user can find out whether the "Detect join condition automatically" option is enabled or not by checking **[Database] > [General]** tab of **[option]** dialog.

Adding fields

Drag & drop your fields from tables in the table area to the the field area, or just click the fields of tables. The selected fields will be added in the field area. Multiple fields can be added at once using <Shift> or <Ctrl> key.

Deleting fields

Select fields to remove in the field area and push <Delete> key.

Sets the sort order for each fields using the query wizard. Click **[Query] > [Query Wizard]** in the menubar, or select **[Query Wizard]** from the right mouse click popup menu on the any area of the query window.

Select [Order By] tab of the query wizard. Add sorting fields from the available fields list to the selected fields list and set sort order (sort type) and sort field order.

The query you are designing is shown in the query area.

Renaming rule for table name :

When your data source is MDB or MS-SQL, if a table name includes blank or number, the table name is renamed in your SQL statement.

Table name includes blank : ex) Car Sales -> [Car Sales].fieldname
Table name begins with a number : ex) 9Sales -> [9Sales].fieldname
Table name is the same as field name : ex) Sales.Sales -> [Sales].Sales

To execute SQL statement, select **[Run] > [Run Query]** from the menubar or click the **[Run Query icon]** on the toolbar or push the <F5> key. If no SQL error, the result will be shown in the [Result] tab.

Multiple datasets can have master-detail relationship between them.
Let's take an example:
Master set : MemberGrade
Detail set : Customer

Create a dataset having the grouping field "customer.member_card" using the query wizard.

Created query would be :

```
SELECT customer.member_card
FROM customer
GROUP BY customer.member_card
```

Continue with SQL to select customer information by customer.member_card :

Our query would be :

```
SELECT customer.customer_id, customer.fname, customer.lname, customer.phone1,
customer.gender
FROM customer
WHERE customer.member_card = '#MemberGrade.member_card#'
```

Select dataset "customer" in the data window for the detail set. And set the "Master Set" property value to "MemberGrade".

Dynamic query

Dynamic Query is a query that some part of the statement such as field definitions is to be determined dynamically at runtime. This is used when it is not allowed to determine fields specification at design time as the fields dynamically change. Dynamic query also enables the end user to manually select the fields and conditions of the query to run.

Dynamic dataset

Dynamic dataset is a 2nd dataset which is made from data sources of database, UDS, FILE/HTTP or XML DTD Store to allow the user to create Dynamic Query Reports.

Instead of storing fields information of the dynamic query, Dynamic Data set stores all query results with three attributes ROWIDX, COLUMN and VALUE. At run time, dynamic field information is retrieved from these three attributes and constructs the dynamic query.

Therefore, the user can design a report using these three attributes of Dynamic Data set, and make the report to be expanded with dynamically created query results at runtime.

The definition of three attributes of Dynamic Data set is as follows:

ROWIDX : row index on the dataset

COLUMN : field name in the dataset

VALUE : value of the field

For example, The dynamic data set "table2" for the source data set "table1" have three fields of ROWIDX, COLUMN, VALUE. All fields of all rows are saved in the dynamic dataset "table2"

■ table1 – table for the source dataset

Region	Sales2000	Sales2001
Canada	157000	890235
East	5234991	5432123

Middle West	1234789	1123454
South	4563300	4992100
West	3551090	3901234

■ table2 – table for the dynamic dataset

ROWIDX	COLUMN	VALUE
0	Region	Canada
0	Sales2000	257000
0	Sales2001	890235
1	Region	East
1	Sales2000	5234991
1	Sales2001	5432123
2	Region	Middle West
2	Sales2000	1234789
2	Sales2001	1123454
3	Region	South
3	Sales2000	4563300
3	Sales2001	4992100
4	Region	West
4	Sales2000	3551090
4	Sales2002	3901234

Add dynamic dataset

Select **[Add Store...]** from the right mouse click popup menu on the [DATABASE] in the data window. Add database store using the [Add database Connection Information] dialog.

Select **[Add Dynamic Dataset...]** from the right mouse click popup menu on the database store you have created.

Enter Dynamic dataset name and Starting row number in the [Add Dynamic Dataset] dialog.

Design the query using the query window in the Designer Mode or Manual mode.

Run the query by selecting **[Run] > [Run Query]** from the menubar or clicking the **[Run Query icon]** on the toolbar and check fields information of the dynamic dataset.

Select **[Add Multi Dataset...]** from the right mouse click popup menu on the database store you have created.

Enter Multi Dataset Name in the [Add Multi Dataset] dialog.

Design the query using the query window in the Manual mode.

Query example :

```
SELECT * FROM Customers
SELECT * FROM Employees
```

To execute SQL statement, select **[Run] > [Run Query]** from the menubar or click the **[Run Query icon]** on the toolbar.

Writing multi-dataset query

Enter procedure or multiple queries in the query window of the multi dataset.

- method 1 : Separate queries by a blank.

```
SELECT * FROM Customers SELECT * FROM Employees
```

- method 2 : Separate queries by a newline.

```
SELECT * FROM Customers
SELECT * FROM Employees
```

The master set property of the sub dataset is not configurable and one of the multi dataset is inherited. Multi dataset is supported only in the manual mode for MSSQL and SYBASE.

Multiple sub datasets can be created using procedures.

Select **[Add Store...]** from the right mouse click popup menu on the [DATABASE] in the data window. Add database store using the [Add database Connection Information] dialog.

Select **[Add Procedure Dataset...]** from the right mouse click popup menu on the database store you have created.

Enter your dataset name, select schema, and click **[Search]** in [Stored Procedure] dialog. Select your procedure from the result list and click **[Next].**

Enter value of the input parameter, select the output parameter and click **[Finish]**. The dataset appears under store.

Naming rule of sub dataset :

The sub dataset name of the procedure dataset is automatically set to the procedure dataset name with appendix "_#" where "#" represents a serial number. For example, if set the procedure dataset name to "MULTISET" in the [Stored Procedure] dialog, The sub dataset name will be "MULTISET_1", "MULTISET_2", etc. To change the dataset name, select **[Modify...]** from the right mouse click popup menu on the dataset.

Auto mapping :

Click the [Auto mapping] automatically set the input parameter to the value in the form of "#OZParam.parametername#".

Input, Output :

	If you want to set the parameters of the procedure to enter the parameter values in the Input option. The parameter values can be entered directly, or you can set a user-defined parameter. If the "Apply Null" is True, you can not directly enter the parameter value. The parameters to get the procedure execution result is set in the Output option.

Select **[Add Parameter Field]** from the right mouse click popup menu on the [USER-DEFINED PARAMETER] in the data window.

Enter your parameter names in the "Member" list and configure properties in the property window in [USER-DEFINED PARAMETER] dialog. User defined parameters appear in the data window.

Dialog Options Description

Name : parameter name

Type : parameter data type

Value : default value of the parameter

Session Key : set parameter value using server session instead of viewer parameter.

Description : parameter description

If configure conditions in a query using user defined parameters, the parameter input dialog will appears at query runtime. If a parameter value is defined in the [USER-DEFINED PARAMETER] dialog, the parameter input dialog will show the value.

Calculated fields can be added to dataset of data source such as database, UDS, XML, CSV, Dynamic and SAP dataset.

Select **[Add Calculated Field]** from the right mouse click popup menu on the dataset under [DATABASE] in the data window.

Enter "Field Name" and check "Script Type" in the [Add Calculated Field] dialog.

The calculated field appears in the data window.

If configure conditions in a query using calculated fields, the parameter input dialog will show empty calculated field value at query runtime.

The user can add an ODI file from the server repository or local work folder to the current ODI file.

Adding ODI dataset

Select **[Add...]** from the right mouse click popup menu on the [INCLUDE ODI] in the data window.

Select the ODI file from the Repository File Window. Enter a field name for "Include field name" option in [Add ODI File] dialog.

The store, datasets, fields, and user defined parameters of the ODI file are added to the INCLUDE field "include_1", but cannot be modified.

Deleting ODI dataset

Select **[Delete]** from the right mouse click popup menu on the INCLUDE field to remove.

Click **[Yes]** in the dialog asking confirmation of removing the INCLUDE field.

Restrictions

When adding a new ODI file to the current ODI file that already have an Include field:

- An ODI of which data source is the same as the current ODI cannot be added.
- An ODI that have the same store name, dataset name, field name with the current ODI cannot be added.
- When adding B.odi including A.odi to C.odi, only the B.odi without the A.odi will be added to C.odi.

The compiled SQL uses "PreparedStatement"and enabled by setting the "Compiled SQL Statement" dataset property to True. The compiled SQL should be supported by each JDBC database.

The way of compiled SQL design with OZ parameter depends on the value of "use script" and "Compiled SQL Statement" dataset property.

Use script = "False", Compiled SQL Statement = "True"/"False"

Use #OZParam.parametername# or '#OZParam.parametername#' as a variable in the SQL statement.

- numeric OZ parameter

```
SELECT * FROM tablename
WHERE fieldname = #OZParam.parametername#
```

- non-numeric OZ parameter

```
SELECT * FROM tablename
WHERE fieldname = '#OZParam.parametername#'
```

Use script = "True", Compiled SQL Statement = "True"

Use #OZParam.parametername# or '#OZParam.parametername#' in the SQL statement text string.

- numeric OZ parameter

```
string sql;
sql = "SELECT * FROM tablename WHERE fieldname = #OZParam.parametername#";
setQueryString(sql);
```

- non-numeric OZ parameter

```
string sql;
sql = "SELECT * FROM tablename WHERE fieldname = '#OZParam.parametername#' ";
setQueryString(sql);
```

Use script = "True", Compiled SQL Statement = "False"

Use #OZParam.parametername# or '#OZParam.parametername#' as a variable in the SQL statement. For non-numeric OZ parameter, the single quotation mark should be included in the double quotation marks.

- numeric OZ parameter

```
string sql;
sql = "SELECT * FROM tablename WHERE fieldname = " + #OZParam.parametername#;
setQueryString(sql);
```

- non-numeric OZ parameter

```
string sql;
sql = "SELECT * FROM tablename WHERE fieldname = " + "'" +
#OZParam.parametername# + "'";
setQueryString(sql);
```

This chapter describes the properties and usage of OZ data tree, data store, dataset, data field, and user defined parameter.

The OZ data tree in the OZ data window shows data stores, datasets, data fields, and user defined parameters in a tree format.

The OZ data tree component has following properties:

Name	Description
Ignore Null Data	Indicates whether to ignore Null Data
User-Defined Parameter Mark	Marker that represent the user-defined parameter
ConcurrentFetchSize	Fetch buffer size in bytes for concurrent binding
Connection Position	Server location to connect
Allow DataService	Indicates whether to allow data service
Display zero before the decimal point	Indicates whether to display zero before the decimal point for the value smaller than 1
Ignore Script Error	Indicates whether to not show the script error message when running a query

Indicates whether to ignore Null Data

Values :

True	When the data type is a character type, it is handled as "". When it is a number type, it is handled as 0
False	Treated as Null data

Reference :

In the Query Designer, as well as Report, to use null to the null data, sets the property value to False. To use null to the "" or 0, sets the property value to True.

To use null in Report, sets the "Ignore Null Data" of "OZ data tree" to False in Query Designer and sets the "Ignore Null Data" of "Data Tree" to False in Report Designer.

Marker that represent the user-defined parameter

Reference :

The default marker is "#".

If the marker is "#", a parameter is represented like "#OZParam.parametername#".

For the manual design mode, if "Use Script" dataset property value is "True", only the default maker(#) is available.

The marker should be one byte special character and alphanumeric characters are not allowed.
The alphabetic, numeric, or 1byte more characters can not be used as a marker.
If the data store type is RDB, the "^" and the characters used in the DB connection information can not be used as a marker.
Also, if the name of user-defined parameter is included the marker, occurs erros when query statement is executed.
Therefore, please be careful when setting the marker.

See also :

OZParamMark

Fetch buffer size in bytes for concurrent binding

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server.

See also :

ConcurrentFetchSize

Server location to connect

Values :

Default	Connect to the server defined by "Connection Position" option (File> Options > Environment> Server Setting) (default)
OZServer	Connect to the current repository server
Local	Connect to the local server

Reference :

The server location to connect depends on the "Connection Position" property value of the data tree and "Connection Position" option value of the Query designer environment setup option.

Option	Property	Description
OpenType	Default	Depends on the location of the current open ODI file (i.e. local server or repository server)
	OZServer	The current repository server. If no repository server is connected, connect to the local server.

	Local	Connect to the local server.
Local	Default	Connect to the local server.
	OZServer	The current repository server. If no repository server is connected, connect to the local server.
	Local	Connect to the local server.
OZServer	Default	The current repository server. If no repository server is connected, connect to the local server.
	OZServer	The current repository server. If no repository server is connected, connect to the local server.
	Local	Connect to the local server.
ForceLocal	Connect to the local server.	
ForceOZServer	The current repository server. If no repository server is connected, connect to the local server.	
ForceOpenType	Depends on the location of the current open ODI file (i.e. local server or repository server)	

Indicates whether to allow data service

Values :

Default	According to the dataservice_defaultType option of ozservice.properties file in OZ Server (default)
True	Allow data service

False	Refuse data service

Indicates whether to display zero before the decimal point for the value smaller than 1

Values :

Default	Depends on the property value of the DB server (default)
True	Display zero before the decimal point
False	Do not display zero before the decimal point

Reference :

This property applies only for Numeric or Decimal data type.

See also :

ShowZeroDecimalPoint

Indicates whether to not show the script error message when running a query

Values :

True	Ignore and not display the script error message
False	Display the script error message (default)

Reference :

TABLE OF CONTENTS

Manages data store for the JDBC/ODBC database data source.

Name	Description
Name	Database store name
User Name	Database user name
JDBC Driver	ODBC/JDBC driver name for database connection
DB Alias	Indicates whether to get the database connection information from the DB Alias File
Alias Name	The database connection alias name to get from the database alias file
DB Alias Filename	Includes database connection alias definitions

Auto Commit	Indicates whether to allow auto commit
Vendor	Database vendor name
DataAction Class	Name of the data action class
DataAction Initialization Parameter	Parameter to initialize data action

Database store name

See also :

GetName

Database user name

ODBC/JDBC driver name for database connection

Indicates whether to get the database connection information from the DB Alias File

Values :

True	Get the database connection information from the DB Alias File
False	Not get the database connection information from the DB Alias File

See also :

UseConnectionPoolAlias

The database connection alias name to get from the database alias file

Reference :

To use a database connection alias name on the server side, the alias must be defined in the db.properties file on the server.

See also :

ConnectionPoolAlias

This file includes database connection alias definitions

Indicates whether to allow auto commit

Values :

True	Allow auto commit
False	Not allow auto commit

Reference :

Generally, JDBC supports auto commit. But there are some restrictions according to user environments.
For example, when Oracle connects to DB2, set as below:

JDBC	**OZ**
com.setAutoCommit(false)	auto commit=false
com.preparedStatement(sqlstr)	Compiled SQL=true

For more information, refer to the JDBC documents provided by the database vendor.

See also :

AutoCommit

Database vendor name

See also :

VendorName

Name of the data action class

Parameter to initialize data action

UDS(User Data Store) manages data store for non-database data sources like data files (TXT, CSV, XML) or user application data (EJB, Servlet, ASP, JSP, Applet).

Name	Description
Name	User Data Store name
Class Name	UDS class file name
Define field information manually	Allow Report Designer to add field definition

User Data Store name

See also :

GetName

UDS class file name

See also :

ClassName

Allow Report Designer to add field definition (name and type)

Values :

True	Allow
False	Not allow

Group data store manages data store for the original data source like a database or UDS and can configure master-detail relationship between source dataset and group data store.

Name	Description
Name	Group data store name
Source Dataset	Original dataset of the group data store

Group data store name

See also :

GetName

Original dataset of the group data store

See also :

SourceDataSetName

SOAP(Simple Object Access Protocol) store manages data store for the web service data source.

Name	Description
Name	SOAP store name

SOAP store name

See also :

GetName

ADO DataSet web service store manages data store for the ADO DataSet object data source.

Name	Description
Name	ADO DataSet web service store name

ADO DataSet web service store name

XML data store manages data store for the data source of XML data with DTD/XSD.

Name	Description
Name	XML store name

XML store name

See also :

GetName

SAP Data Store manages data store for the result data of SAP R/3 RFC(Remote Function Call).

Name	Description
Client Name	Client number to login
Language	The language to use

Host Name	SAP server host name or IP Address
System Number	SAP server system number
DataAction Class	Name of the data action class
DataAction Initialization Parameter	Parameter to initialize data action
SID	SID of SAP Netweaver Server connection pool

Client number to login

See also :

ClientName

The language to use

See also :

Language

SAP server host name or IP Address

See also :

HostName

SAP server system number

See also :

SystemNumber

Name of the data action class

Parameter to initialize data action

SID of SAP Netweaver Server connection pool

See also :

SID

Clear Quest Store manages data store for the result data of IBM Rational Clear Quest.

Name	Description
Name	CLEAR QUEST Store name

CLEAR QUEST Store name

TABLE OF CONTENTS

Manages the dataset created in the database store using SQL.

Name	Description
Name	Query dataset name
SQL Statement	SQL statement to execute
Master Set	Master set name
Maximum Rows	Maximum row count of dataset
Compiled SQL Statement	Allow compiled SQL statement
JDBC Fetch Row	Row count of the result dataset by one transmission

Script	Allow scripted SQL
Hidden	Whether to hide the data set
ConcurrentFetchSize	Fetch buffer size for concurrent binding
ConcurrentFirstRow	First fetch row count for concurrent binding
Send data before binding	Flush output stream buffer to the client before dataset binding
Send data after binding	Flush output stream buffer to the client after dataset binding
Query Timeout	Query execution timeout
Ignore Query Errors	Whether to ignore query errors
Description	Dataset description

Database Store dataset name.

See also :

GetName

SQL statement to execute.

See also :

QueryString

Master set name.

Reference :

Specify the master set of the current dataset.

See also :

GetMasterDataSetName

Maximum row count of dataset.

Reference :

Specify the maximum row count of the result dataset.

It should be in range of 0 ~ 2147483647.

See also :

MaxRowCount

Allow compiled SQL statement.

Values :

True	Allowed
False	Not allowed

Reference :

Using of the compiled SQL improves SQL performance, when repeating the same SQL. Usage of the compiled SQL depends on the database vendors (JDBC database provides preparedstatement).

This function works only when executing the SQL on the server. When Query Designer executes the SQL, even though set to True, the SQL is executed as non-compiled SQL.

See also :

UseCompiledSQL

Row count of the result dataset by one transmission.

See also :

JDBCFetchRow

Allow scripted SQL

Values :

True	Allow
False	Not allow

See also :

UseOZScript

Whether to hide the data set

Values :

True	Hide the data set
False	Display the data set

Reference :

The hidden data sets are not displayed in the data tree of Report Designer.
Also the hidden data sets can not be accessed by JavaScript and the result set is not sent to the viewer.

If you hide the source dataset for grouping in group data store, the oz server does not send the source dataset.

See also :

Hidden

Fetch buffer size in bytes for concurrent binding.

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFetchSize

First fetch row count for concurrent binding.

Reference :

When the result dataset have very large size, you can quickly get the first small part of the dataset.

If set to zero, the count defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFirstRow

Flush output stream buffer to the client before dataset binding.

Values :

True	Flush buffer before dataset binding
False	Not flush buffer before dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataBeforeBinding

Flush output stream buffer to the client after dataset binding

Values :

True	Flush buffer after dataset binding
False	Not flush buffer after dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataAfterBinding

Query execution timeout

Reference :

Set to a value between 0 and 2147483647.

See also :

QueryTimeout

Whether to ignore query errors

Values :

True	Send an empty data set containing field information to the viewer when a query error occurs
False	Do not send data to the viewer when a query error occurs

See also :

IgnoreQueryError

Dataset description.

Manages the dynamic dataset created to be used for a dynamic query.

Name	Description
Name	Query dataset name
Master Set	Master set name
Maximum Rows	Maximum row count of dataset
Starting Row Number	Starting row number
Hidden	Whether to hide the data set
ConcurrentFetchSize	Concurrent type fetch buffer size
ConcurrentFirstRow	First fetch row count for concurrent binding
Compiled SQL Statement (Source Dataset)	Allow compiled SQL statement for the original set
Use Script (Source Set)	Allow scripted SQL for the original set
Send data before binding	Flush output stream buffer to the client before dataset binding
Send data after binding	Flush output stream buffer to the client after dataset binding
Query Timeout	Query execution timeout
Ignore Query Errors	Whether to ignore query errors

Description	Dataset description

Query dataset name.

Master set name.

Reference :

Specify the master set of the current dataset.

Maximum row count of dataset.

Reference :

Specify the maximum row count of the result dataset.

It should be in range of 0 ~ 2147483647.

Starting row number.

Values :

The starting number of ROW INDEX on the dynamic dataset.

Reference :

The default starting number is "0".

It should be in range of 0 ~ 2147483647.

Whether to hide the data set

Values :

True	Hide the data set
False	Display the data set

Reference :

The hidden data sets are not displayed in the data tree of Report Designer.
Also the hidden data sets can not be accessed by JavaScript and the result set is not sent to the viewer.

If you hide the source dataset for grouping in group data store, the oz server does not send the source dataset.

See also :

Hidden

Fetch buffer size in bytes for concurrent binding.

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

First fetch row count for concurrent binding.

Reference :

When the result dataset have very large size, you can quickly get the first small part of the dataset.

If set to zero, the count defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

Allow compiled SQL statement for the original set.

Values :

True	Allowed
False	Not allowed

Reference :

Using the compiled SQL improves SQL performance, when repeating the same SQL. Usage of the compiled SQL depends on the database vendors (JDBC database provides preparedstatement).

This feature works only when executing the SQL on the server. When Query Designer executes the SQL, even though set to True, the SQL is executed as non-compiled SQL.

Allow scripted SQL for the original set

Values :

True	Allowed
False	Not allowed

Flush output stream buffer to the client before dataset binding.

Values :

True	Flush buffer before dataset binding
False	Not flush buffer before dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

Flush output stream buffer to the client after dataset binding

Values :

True	Flush buffer after dataset binding
False	Not flush buffer after dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

Query execution timeout

Reference :

Set to a value between 0 and 2147483647.

See also :

QueryTimeout

Whether to ignore query errors

Values :

True	Send an empty data set containing field information to the viewer when a query error occurs
False	Do not send data to the viewer when a query error occurs

See also :

IgnoreQueryError

Dataset description.

Manages the multi dataset created by multiple query.

Using of the SQL that executes a procedure depends on the database vendors.

Name	Description
Name	Query dataset name
SQL Statement	Query statement to execute
Master Set	Master set name
Compiled SQL Statement	Allow compiled SQL for the original set
Script	Allow SQL in script
Query Timeout	Query execution timeout

Query dataset name

See also :

GetName

Query statement to execute

See also :

QueryString

Master set name

Reference :

Gets or sets the master set name of the dataset

See also :

GetMasterDataSetName

Allow compiled SQL for the original set

Values :

True	Allow
False	Not allow

Reference :

Using the compiled SQL improves SQL performance, when repeating the same SQL. Usage of the compiled SQL depends on the database vendors (JDBC database provides preparedstatement).

This feature works only when executing the SQL on the server. When Query Designer executes the SQL, even though set to True, the SQL is executed as non-compiled SQL.

See also :

UseCompiledSQL

Allow SQL in script.

Values :

True	Allow
False	Not allow

See also :

UseOZScript

Query execution timeout

Reference :

Set to a value between 0 and 2147483647.

See also :

QueryTimeout

Manages the multi dataset created by a dataset procedure.

Name	Description
Name	Query dataset name
Procedure name	procedure name
Procedure type	procedure type
Target parameter	target parameter name
Master Set	master set name
Query Timeout	Query execution timeout

Query dataset name

See also :

GetName

Procedure name

Values :

Gets or sets the procedure name.

See also :

ProcedureName

Procedure type

Values :

Gets or sets the procedure type.

Reference :

The procedure type can be set by a procedure or function.

See also :

ProcedureType

Target parameter name

Values :

Sets the target parameter name to get the result of procedure.

Reference :

Target parameter name is shown only when the parameter type is Cursor, and no name is shown for other parameter types.

See also :

TargetParameter

Master set name

Reference :

Sets the master set name for the current dataset.

See also :

GetMasterDataSetName

Query execution timeout

Reference :

Set to a value between 0 and 2147483647.

See also :

QueryTimeout

Manages the dataset created in the User Data Store using a query.

Name	Description

Name	User Data Store dataset name
Statement	Execution statement
Master Set	Master set name
Maximum Rows	Maximum row count of dataset
Script	Allow scripted SQL
Hidden	Whether to hide the data set
ConcurrentFetchSize	Fetch buffer size for concurrent binding
ConcurrentFirstRow	First fetch row count for concurrent binding
Send data before binding	Flush output stream buffer to the client before dataset binding
Send data after binding	Flush output stream buffer to the client after dataset binding
Description	Dataset description

User data Store dataset name.

See also :

GetName

Execution statement.

Reference :

Sets the execution statement to get data from UDS class.

The statement will be passed to the UDS class as an argument.

See also :

Command

Master set name.

Reference :

Specify the master set of the current dataset.

See also :

GetMasterDataSetName

Maximum row count of dataset.

Reference :

Specify the maximum row count of the result dataset.

It should be in range of 0 ~ 2147483647.

Allow scripted SQL.

Values :

True	Allow
False	Not allow

See also :

UseOZScript

Whether to hide the data set

Values :

True	Hide the data set
False	Display the data set

Reference :

The hidden data sets are not displayed in the data tree of Report Designer.
Also the hidden data sets can not be accessed by JavaScript and the result set is not sent to the viewer.

If you hide the source dataset for grouping in group data store, the oz server does not send the source dataset.

See also :

Hidden

Fetch buffer size in bytes for concurrent binding.

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFetchSize

First fetch row count for concurrent binding.

Reference :

When the result dataset have very large size, you can quickly get the first small part of the dataset.

If set to zero, the count defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFirstRow

Flush output stream buffer to the client before dataset binding.

Values :

True	Flush buffer before dataset binding
False	Not flush buffer before dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataBeforeBinding

Flush output stream buffer to the client after dataset binding.

Values :

True	Flush buffer after dataset binding
False	Not flush buffer after dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataAfterBinding

Dataset description.

Manages the dataset created in the Group Data Store using a query.

Name	Description
Name	Group Store dataset name
Statement	Execution statement
Master Set	Master set name
Consume Row	Splits the original dataset into multiple datasets.
Maximum Rows	Maximum row count of dataset
Hidden	Whether to hide the data set
ConcurrentFetchSize	Fetch buffer size for concurrent binding
ConcurrentFirstRow	First fetch row count for concurrent binding
Send data before binding	Flush output stream buffer to the client before dataset binding
Send data after binding	Flush output stream buffer to the client after dataset binding
Description	Dataset description

Group Store dataset name.

See also :

GetName

Execution statement.

Reference :

The execution statement to create the group data.

See also :

Command

Master set name.

Reference :

Specify the master set of the current dataset.

See also :

GetMasterDataSetName

Splits the original dataset into multiple datasets.

Values :

True	Split
False	No split

Reference :

When data splitting of a row is finished and getting next row, "True" must be set to complete group store binding. Normally, the dataset of the lowest level in a group store might be set to "True".

Maximum row count of dataset.

Reference :

Specify the maximum row count of the result dataset.

It should be in range of 0 ~ 2147483647.

See also :

MaxRowCount

Whether to hide the data set

Values :

True	Hide the data set
False	Display the data set

Reference :

The hidden data sets are not displayed in the data tree of Report Designer.
Also the hidden data sets can not be accessed by JavaScript and the result set is not sent to the viewer.

If you hide the source dataset for grouping in group data store, the oz server does not send the source dataset.

See also :

Hidden

Fetch buffer size in bytes for concurrent binding.

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFetchSize

First fetch row count for concurrent binding.

Reference :

When the result dataset have very large size, you can quickly get the first small part of the dataset.

If set to zero, the count defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFirstRow

Flush output stream buffer to the client before dataset binding.

Values :

True	Flush buffer before dataset binding
False	Not flush buffer before dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataBeforeBinding

Flush output stream buffer to the client after dataset binding.

Values :

True	Flush buffer after dataset binding
False	Not flush buffer after dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataAfterBinding

Dataset description.

Manages the FILE STORE XML dataset that is derived from user application's output such as XML file or text stream.

Name	Description
Name	FILE STORE XML dataset name.
Master Set	Master set name.
Maximum Rows	Maximum row count of dataset.
Dataset Node Name	The dataset node name in XML file.
Recordset Node Name	The recordset node name in XML file.
Hidden	Whether to hide the data set
ConcurrentFetchSize	Fetch buffer size for concurrent binding
ConcurrentFirstRow	First fetch row count for concurrent binding
Send data bofore binding	Flush output stream buffer to the client before dataset binding
Send data after binding	Flush output stream buffer to the client after dataset binding

Description	Dataset description

FILE STORE XML dataset name.

See also :

GetName

Master set name.

Reference :

Specify the master set of the current dataset.

See also :

GetMasterDataSetName

Maximum row count of dataset.

Reference :

Specify the maximum row count of the result dataset.

It should be in range of 0 ~ 2147483647.

See also :

MaxRowCount

The dataset node name in XML file.

See also :

DataSetNodeName

The recordset node name in XML file.

See also :

RecordNodeName

Whether to hide the data set

Values :

True	Hide the data set
False	Display the data set

Reference :

The hidden data sets are not displayed in the data tree of Report Designer.
Also the hidden data sets can not be accessed by JavaScript and the result set is not sent to the viewer.

If you hide the source dataset for grouping in group data store, the oz server does not send the source dataset.

See also :

Hidden

Fetch buffer size in bytes for concurrent binding.

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFetchSize

First fetch row count for concurrent binding.

Reference :

When the result dataset have very large size, you can quickly get the first small part of the dataset.

If set to zero, the count defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFirstRow

Flush output stream buffer to the client before dataset binding.

Values :

True	Flush buffer before dataset binding
False	Not flush buffer before dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataBeforeBinding

Flush output stream buffer to the client after dataset binding

Values :

True	Flush buffer after dataset binding
False	Not flush buffer after dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataAfterBinding

Dataset description.

Manages the FILE STORE CSV dataset that is derived from user application's output such as CSV file or text stream.

Name	Description
Name	FILE STORE CSV dataset name.
Master Set	Master set name.
Maximum Rows	Maximum row count of dataset.
Field Delimiter	Field delimiter
Line Delimiter	Line delimiter
Character Set	Character set of CSV file
Null Mark	Null marker
Hidden	Whether to hide the data set
ConcurrentFetchSize	Fetch buffer size for concurrent binding
ConcurrentFirstRow	First fetch row count for concurrent binding
Send data before binding	Flush output stream buffer to the client before dataset binding
Send data after binding	Flush output stream buffer to the client after dataset binding

Description	Dataset description

FILE STORE CSV dataset name.

See also :

GetName

Master set name.

Reference :

Sets the master set of the current dataset.

See also :

GetMasterDataSetName

Maximum row count of dataset.

Reference :

Specify the maximum row count of the result dataset.

It should be in range of 0 ~ 2147483647.

See also :

MaxRowCount

Field delimiter

Reference :

Define a character as the field delimiter.

See also :

ColumnDelimiter

Line delimiter.

Reference :

Define a character as the line delimiter.

Reference :

The default value is LINEFEED, the new line character(\n).

The line delimiter is case sensitive.

Old version of Query Designer provides only the LINEFEED as the line delimiter.

Character set of CSV file.

See also :

CharacterSet

Null marker represents the null value.

See also :

NullMark

Whether to hide the data set

Values :

True	Hide the data set
False	Display the data set

Reference :

The hidden data sets are not displayed in the data tree of Report Designer.
Also the hidden data sets can not be accessed by JavaScript and the result set is not sent to the viewer.

If you hide the source dataset for grouping in group data store, the oz server does not send the source dataset.

See also :

Hidden

Fetch buffer size in bytes for concurrent binding.

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFetchSize

First fetch row count for concurrent binding.

Reference :

When the result dataset have very large size, you can quickly get the first small part of the dataset.

If set to zero, the count defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFirstRow

Flush output stream buffer to the client before dataset binding.

Values :

True	Flush buffer before dataset binding
False	Not flush buffer before dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataBeforeBinding

Flush output stream buffer to the client after dataset binding

Values :

True	Flush buffer after dataset binding
False	Not flush buffer after dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataAfterBinding

Dataset description.

Manages the HTTP STORE XML dataset that is derived from user application's output such as XML file or text stream.

Name	Description
Name	HTTP STORE XML dataset name
Master Set	Master set name.
Maximum Rows	Maximum row count of dataset.
Dataset Node Name	The dataset node name in XML file.
Recordset Node Name	The recordset node name in XML file.
Hidden	Whether to hide the data set
ConcurrentFetchSize	Fetch buffer size for concurrent binding
ConcurrentFirstRow	First fetch row count for concurrent binding

Send data before binding	Flush output stream buffer to the client before dataset binding
Send data after binding	Flush output stream buffer to the client after dataset binding
Description	Dataset description

HTTP STORE XML dataset name

Master set name.

Reference :

Sets the master set of the current dataset.

Maximum row count of dataset.

Reference :

Specify the maximum row count of the result dataset.

It should be in range of 0 ~ 2147483647.

See also :

MaxRowCount

The dataset node name in XML file.

See also :

DataSetNodeName

The recordset node name in XML file.

See also :

RecordNodeName

Whether to hide the data set

Values :

True	Hide the data set
False	Display the data set

Reference :

The hidden data sets are not displayed in the data tree of Report Designer.
Also the hidden data sets can not be accessed by JavaScript and the result set is not sent to the viewer.

If you hide the source dataset for grouping in group data store, the oz server does not send the source dataset.

See also :

Hidden

Fetch buffer size in bytes for concurrent binding.

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFetchSize

First fetch row count for concurrent binding.

Reference :

When the result dataset have very large size, you can quickly get the first small part of the dataset.

If set to zero, the count defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFirstRow

Flush output stream buffer to the client before dataset binding.

Values :

True	Flush buffer before dataset binding
False	Not flush buffer before dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataBeforeBinding

Flush output stream buffer to the client after dataset binding

Values :

True	Flush buffer after dataset binding
False	Not flush buffer after dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataAfterBinding

Dataset description.

Manages the HTTP STORE CSV dataset that is derived from user application's output such as CSV file or text stream.

Name	Description
Name	HTTP STORE CSV dataset name.
Master Set	Master set name.
Maximum Rows	Maximum row count of dataset.
Field Delimiter	Field delimiter
Line Delimiter	Line delimiter
Character Set	Character set of CSV file
Null Mark	Null marker
Hidden	Whether to hide the data set
ConcurrentFetchSize	Fetch buffer size for concurrent binding
ConcurrentFirstRow	First fetch row count for concurrent binding
Send data before binding	Flush output stream buffer to the client before dataset binding
Send data after binding	Flush output stream buffer to the client after dataset binding
Description	Dataset description

HTTP STORE CSV dataset name.

Master set name.

Reference :

Sets the master set of the current dataset.

Maximum row count of dataset.

Reference :

Specify the maximum row count of the result dataset.

It should be in range of 0 ~ 2147483647.

See also :

MaxRowCount

Field delimiter

Reference :

Define a character as the field delimiter.

See also :

ColumnDelimiter

Line delimiter.

Reference :

Define a character as the line delimiter.

Reference :

The default value is LINEFEED, the new line character(\n).

The line delimiter is case sensitive.

Old version of Query Designer provides only the LINEFEED as the line delimiter.

Character set of CSV file.

See also :

CharacterSet

Null marker represents the null value.

See also :

NullMark

Whether to hide the data set

Values :

True	Hide the data set
False	Display the data set

Reference :

The hidden data sets are not displayed in the data tree of Report Designer.
Also the hidden data sets can not be accessed by JavaScript and the result set is not sent to the viewer.

If you hide the source dataset for grouping in group data store, the oz server does not send the source dataset.

See also :

Hidden

Fetch buffer size in bytes for concurrent binding.

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFetchSize

First fetch row count for concurrent binding.

Reference :

When the result dataset have very large size, you can quickly get the first small part of the dataset.

If set to zero, the count defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFirstRow

Flush output stream buffer to the client before dataset binding.

Values :

True	Flush buffer before dataset binding
False	Not flush buffer before dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataBeforeBinding

Flush output stream buffer to the client after dataset binding

Values :

True	Flush buffer after dataset binding
False	Not flush buffer after dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataAfterBinding

Dataset description.

Manages the dataset created in SOAP store. SOAP STORE dataset is made from the web service data source.

Name	Description
Name	SOAP STORE dataset name.
Master Set	Master set name.
Hidden	Whether to hide the data set
ConcurrentFetchSize	Fetch buffer size for concurrent binding
ConcurrentFirstRow	First fetch row count for concurrent binding
Send data before binding	Flush output stream buffer to the client before dataset binding
Send data after binding	Flush output stream buffer to the client after dataset binding
Base XPath	Base XML Path
Description	Dataset description

SOAP STORE dataset name.

Master set name.

Reference :

Sets the master set of the current dataset.

Whether to hide the data set

Values :

True	Hide the data set
False	Display the data set

Reference :

The hidden data sets are not displayed in the data tree of Report Designer.
Also the hidden data sets can not be accessed by JavaScript and the result set is not sent to the viewer.

If you hide the source dataset for grouping in group data store, the oz server does not send the source dataset.

Fetch buffer size in bytes for concurrent binding.

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

First fetch row count for concurrent binding.

Reference :

When the result dataset have very large size, you can quickly get the first small part of the dataset.

If set to zero, the count defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

Flush output stream buffer to the client before dataset binding.

Values :

True	Flush buffer before dataset binding
False	Not flush buffer before dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

Flush output stream buffer to the client after dataset binding

Values :

True	Flush buffer after dataset binding
False	Not flush buffer after dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

Base XML Path

Dataset description

Manages the dataset created in ADO DataSet Webservice Store. The dataset is made from the ADO DataSet object.

Name	Description
Name	ADO multi-dataset name.
Master Set	Master set name.
EndPoint	End Point
SOAPAction	SOAP Action

ADO multi-dataset name.

Master set name.

Reference :

Sets the master set of the current dataset.

EndPoint

SOAPAction

Manages the dataset created in the XML data store. The dataset is made from the XML data with DTD/XSD.

Name	Description

Name	XML STORE dataset name.
Master Set	Master set name.
Hidden	Whether to hide the data set
ConcurrentFetchSize	Concurrent type fetch buffer size
ConcurrentFirstRow	First fetch row count for concurrent binding
Send data before binding	Flush output stream buffer to the client before dataset binding
Send data after binding	Flush output stream buffer to the client after dataset binding
Description	Dataset description

XML STORE dataset name.

See also :

GetName

Master set name.

Reference :

Sets the master set of the current dataset.

See also :

GetMasterDataSetName

Whether to hide the data set

Values :

True	Hide the data set
False	Display the data set

Reference :

The hidden data sets are not displayed in the data tree of Report Designer.
Also the hidden data sets can not be accessed by JavaScript and the result set is not sent to the viewer.

If you hide the source dataset for grouping in group data store, the oz server does not send the source dataset.

See also :

Hidden

Fetch buffer size in bytes for concurrent binding.

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFetchSize

First fetch row count for concurrent binding.

Reference :

When the result dataset have very large size, you can quickly get the first small part of the dataset.

If set to zero, the count defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFirstRow

Flush output stream buffer to the client before dataset binding.

Values :

True	Flush buffer before dataset binding
False	Not flush buffer before dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataBeforeBinding

Flush output stream buffer to the client after dataset binding

Values :

True	Flush buffer after dataset binding
False	Not flush buffer after dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataAfterBinding

Dataset description.

Manages the dataset created in the SAP data store. The dataset is made from the result data of SAP R/3 RFC.

Name	Description
Name	SAP STORE dataset name
Dataset	Name of result dataset created by SAP R/3 RFC.
Hidden	Whether to hide the data set
ConcurrentFetchSize	Concurrent type fetch buffer size
ConcurrentFirstSize	First fetch row count for concurrent binding
Send data before binding	Flush output stream buffer to the client before dataset binding
Send data after binding	Flush output stream buffer to the client after dataset binding
Description	Dataset description

SAP STORE dataset name

See also :

GetName

Name of result dataset created by SAP R/3 RFC.

Whether to hide the data set

Values :

True	Hide the data set
False	Display the data set

Reference :

The hidden data sets are not displayed in the data tree of Report Designer.
Also the hidden data sets can not be accessed by JavaScript and the result set is not sent to the viewer.

If you hide the source dataset for grouping in group data store, the oz server does not send the source dataset.

See also :

Hidden

Fetch buffer size in bytes for concurrent binding.

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFetchSize

First fetch row count for concurrent binding.

Reference :

When the result dataset have very large size, you can quickly get the first small part of the dataset.

If set to zero, the count defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

See also :

ConcurrentFirstRow

Flush output stream buffer to the client before dataset binding.

Values :

True	Flush buffer before dataset binding
False	Not flush buffer before dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataBeforeBinding

Flush output stream buffer to the client after dataset binding

Values :

True	Flush buffer after dataset binding
False	Not flush buffer after dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

See also :

FlushDataAfterBinding

Dataset description.

Manages the dataset created in the Clear Quest data store. The dataset is made from the result data of IBM Rational Clear Quest.

Name	Description
Name	CLEAR QUEST STORE dataset name
Type	CLEAR QUEST STORE data type
Master Set	Master set name
Maximum Rows	Maximum row count of dataset.
Use Script	Allow scripted SQL
Hidden	Whether to hide the data set
ConcurrentFetchSize	Concurrent type fetch buffer size
ConcurrentFirst Row	First fetch row count for concurrent binding

Send data before binding	Flush output stream buffer to the client before dataset binding
Send data after binding	Flush output stream buffer to the client after dataset binding
Description	Dataset description

CLEAR QUEST STORE dataset name.

CLEAR QUEST STORE data type.

Master set name.

Reference :

Sets the master set of the current dataset.

Maximum row count of dataset.

Reference :

Specify the maximum row count of the result dataset.

It should be in range of 0 ~ 2147483647.

Allow scripted SQL

Values :

True	Allow
False	Not allow

Whether to hide the data set

Values :

True	Hide the data set
False	Display the data set

Reference :

The hidden data sets are not displayed in the data tree of Report Designer.
Also the hidden data sets can not be accessed by JavaScript and the result set is not sent to the viewer.

If you hide the source dataset for grouping in group data store, the oz server does not send the source dataset.

Fetch buffer size in bytes for concurrent binding.

Reference :

Size of the result dataset to receive from the server at one time.

If set to zero, the size defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

First fetch row count for concurrent binding.

Reference :

When the result dataset have very large size, you can quickly get the first small part of the dataset.

If set to zero, the count defined by the server applies.

This property does not works when running a query in Query Designer, but applies when the Report viewer receives dataset from the server. The fetch unit setup parameter of the viewer should be set to the dataset unit.

Flush output stream buffer to the client before dataset binding.

Values :

True	Flush buffer before dataset binding
False	Not flush buffer before dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

Flush output stream buffer to the client after dataset binding

Values :

True	Flush buffer after dataset binding
False	Not flush buffer after dataset binding

Reference :

This function works only when the fetch type is "concurrent" and the fetch unit is "data module".

Dataset description.

TABLE OF CONTENTS

Manages fields information of the result dataset.

Name	Description
Name	Name of the data field
Type	Type of the data field
Decryption	Indicates whether to decrypt the data field
Description	Description of the data field

Name of the data field.

See also :

GetFieldName

Type of the data field.

See also :

GetFieldType

Indicates whether to decrypt the data field.

Values :

True	Decrypt
False	Do not decrypt (default)

Reference :

This function is supported only by the Java server, and the decryption status can be checked in the OZ viewer.

To use this function, you must set the values of datafield_usersecurity.Active and datafield_usersecurity.Class in the config/spmgr.properties file of the Java server.

See also :

IsDecrypt

Description of the data field.

Manages fields information of the SOAP result dataset.

Name	Description
Name	Name of the SOAP data field
Type	Type of the SOAP data field
XPath	Base XPath of the SOAP data field

Decryption	Indicates whether to decrypt the SOAP data field
Description	Description of the SOAP data field

Name of the SOAP data field.

Type of the SOAP data field.

Base XPath of the SOAP data field.

Indicates whether to decrypt the SOAP data field.

Values :

Tr	Decrypt

ue	
False	Do not decrypt (default)

Reference :

This function is supported only by the Java server, and the decryption status can be checked in the OZ viewer.

To use this function, you must set the values of datafield_usersecurity.Active and datafield_usersecurity.Class in the config/spmgr.properties file of the Java server.

Description of the SOAP data field.

Manages fields information of the XML result dataset.

Name	Description

Name	Name of the XML data field
Type	Type of the XML data field
XPath	Base XPath of the XML data field
Decryption	Indicates whether to decrypt the XML data field
Description	Description of the XML data field

Name of the XML data field.

See also :

GetFieldName

Type of the XML data field.

See also :

GetFieldType

Base XPath of the XML data field.

See also :

GetXPath

Indicates whether to decrypt the XML data field.

Values :

True	Decrypt
False	Do not decrypt (default)

Reference :

This function is supported only by the Java server, and the decryption status can be checked in the OZ viewer.

To use this function, you must set the values of datafield_usersecurity.Active and datafield_usersecurity.Class in the config/spmgr.properties file of the Java server.

See also :

IsDecrypt

Description of the XML data field.

Manages user defined parameters.

Name	Description
Name	User defined parameter name
Type	User defined parameter type
Value	User defined parameter value
Encryption	Indicates whether to encrypt the User defined parameter
Session Key	User defined parameter's Session Key
Description	User defined parameter description

Name of the user defined parameter variable.

Type of the user defined parameter variable.

Value of the user defined parameter variable.

Reference :

Set the default value of the user defined parameter variable.

Indicates whether to encrypt the User defined parameter.

Values :

True	Encrypt
False	Do not encrypt (default)

Reference :

To use this function, you must set the values of odiparam_usersecurity.Active and odiparam_usersecurity.Class in the config/spmgr.properties file of the Java server.

Session Key of the user defined parameter.

Reference :

This property is supported only for OZ Server Servlet type.

Description of the user defined parameter.

Manages Include fields those are included from the ODI file on server repository or LocalExplorer.

Name	Description
Name	Include field name.
OZURL	Path of the ODI file included

Include field name.

Path of the ODI file included.

Reference :

OZURL is an URL path representing the category and file name of the Included ODI file.

If the Included ODI file is located on the server repository, OZURL is the path for [Repository Root]. If the Included ODI file is not located on the server repository, OZURL is the path for [LocalExplorer].

Ex) ozp:///CategoryName/FileName

This chapter explains environments setup, usage of server repository and so on.

Explains how to set up query designer's environments.

Working folder is very important for file sharing by repository. It is an local working folder to which
Repository client copies file from repository server and saves to an local working folder. That local working folder is the working folder and is very important for file sharing based on repository.

Default Working folder

If the first time you run the OZ Query Designer, working folder path is the OZRepository in the My Documents folder of user accounts. If the OZRepository folder does not exist, it will be created automatically.

Set Working Folder menu

Select **[Set Working Folder]** from the right click popup menu on a folder which is to be working folder in repository window. The selected folder becomes working folder.

If set working folder using local explorer, the working folder will be shown in "Working Folder [path]" format.

If the selected folder has already registered in Working folder manager, both name and path of working folder are shown.
For example, when "C:/Repository" path has already registered as named "Working folder1" in Working folder manager, if set working folder in repository window, working folder name is shown in "(Working Folder1) [C:\Repository]" format with its path.

Working folder Manager

The working folder manager registers multiple working folders from the local explorer and manages them.

Select **[Working folder manager]** from the right click popup menu on the working folder in repository window.

If you click the [Import] button in the Working folder manager dialog, import an XML file containing the working folder list information and displays the working folder list. If you click the [Export] button in the working folder list dialog, save working folder list information that is added to the dialog as an XML file.

If you set the working folder in the Working folder manager, working folder is shown in "Working Folder (Name of working folder) [Path of working folder]" format.

Sets basic designer options such as layout, language, etc.

Use Options dialog by selecting **[File] > [Options...]** menu. Click **[Restore]** to return to default settings.

General

Limit max rows of result data : Set the maximum number of query result rows.
"Max rows" option is available only in **[Result]** tap.

Layout

Design Mode :	**Design View :** design view mode **Manual :** manual mode
Save window layout when the application quits :	Save the current window layout when exiting query designer
Reset window layout :	Reset window layout
Save current window layout :	Save the current window layout

Language

Language :	**Korean :** Korean **Japanese :** Japanese **English :** English **Chinese :** Chinese
Reset :	Reset to the language that is currently running.

Save

Autosave :	Automatic backup every given minutes in range of 1 ~ 60.
Encrypt the file while saving :	Encrypt file before saving. For more details, contact OZ technical support.

Server

Local server type :	Type of OZ local server **Java :** Java type **.Net :** .Net type
Data binding server type :	Specify location of OZ server to bind data for a query. **OpenType :** connect to the server that opened ODI file. **Local :** Local server **OZServer :** current OZ server **ForceLoal :** always local server. **ForceOZServer :** always current OZ server **ForceOpenType :** always the server that opened ODI file. Option values except "ForceLocal" can be set to different values according to "Connection Position" property value of OZ Data Tree.

See also :

Server Connection Position

Set options for database and SQL function list.

Use Options dialog launched by selecting **[File] > [Options...]** menu.

General

Detect join condition automatically :	Automatically detect join conditions between tables.
Use table information cached in the local system :	Save in local cache the table information received from the database at fist connection and use table information on local cache from the next. **Confirm when using table information cached in the local system :** show confirm message box when use table information on local cache.
Get table information automatically when connecting to database server :	Get table information whenever connect to database.

SQL Function List

Set up SQL function list file for each database vendor :	Set SQL function list file for each database vendor. **Database vendor :** select a database vendor. **SQL function list file :** SQL function list file

SQL function list file for database is used with query wizard. If not set it, the function names provided by JDBC Driver will be used. but, function name sometimes may not accurate.

example on SQL function list file.

```
ASCII REPLACE
SUM MIN
USER DUMP
SYSDATE LAST_DAY NEXT_DAY
```

Each function is separated by blank character(Space or Tab).

	The file have four lines and the first line includes string functions, the second line includes math functions, the third line includes system functions, and the fourth line includes time functions.

Set options for query edit window in manual mode, query text window in design view mode, text in script window.

Use Options dialog launched by selecting **[File] > [Options...]** menu.

Environment

Use syntax highlighting : When creating a query in manual mode, display keywords in different colors.

View script help tooltip : Show prototype of a function by tooltip in script window.

Show line numbers : Show line numbers.

Font And Color

Font : font face

Size : font size

Style : font style

Foreground : foreground color

Background : background color

Repository server is an remote storage for various files such as ODI, OZR, OZA, and data file. With Repository server, many developers can share files remotely. Each user can have different file access permission. Version control feature of repository server improves teamwork productivity.

Select **[Open Repository...]** form the right click popup menu on the **[Repository Server]** in the repository window. Double click a server to be connected in server list.

In the Server List dialog, set the server name, type, and connection information, and then click the [Save] button.

If you click the [Import] button in the server list dialog, import an XML file containing the server list information and displays the server list. If you click the [Export] button in the server list dialog, save server list information that is added to the dialog as an XML file.

Open server repository

Explains how to connect to the repository server.

Select [Open Repository...] form the right click popup menu on the [Repository Server] in the repository window. Double click a server to be connected in server list.

Repository server folder and files are appears in the repository window after successful connection.

Close server repository

Explains how to disconnect from the repository server.

Select **[Close Repository...]** form the right click popup menu on the connected repository server in the repository window.

Once connected to the repository server, you can open and edit an ODI file saved on the repository server.

Repository File Window appears by clicking [File] > [Open] menu or the open icon on the toolbar .

Click **[OK]** after selecting a file on the repository server.

Caution for opening file from server

- When you opened a file from the server and saved it on local working folder, if you select **[File]** > **[Save]** or click [save icon], the file will be saved as local file. So, If open the file from server again, the file on server will be opened and might be different from the file saved on local working folder.
- To apply modification of file to repository server, you must check out first, edit the file, and check in last.
- If you have finished editing the file opened from the server, you must check in or cancel checkout.

Add category

Select **[New Category...]** from the right click popup menu on the category to which a new category to be added.

Enter category name in [Add a category] dialog.

File upload

Select the file to upload from working folder and drag & drop to the wanted category in repository server. Item upload dialog appears and shows the item list to be uploaded.

The selected items are uploaded to the designated category in repository server by clicking **[Upload]**.
Select multiple files while pressing **<Ctrl>** or **<Shift>**. Drag & drop the selected files to target category and then Item upload dialog appears and shows the item list to be uploaded. Select items and click **[Upload]**.

File download

Select **[Download...]** from the right click popup menu on the file in repository server to download.

Select items to download from Download Items dialog to working folder in local computer and click **[Download]**.

By checking "Select checked out items" option, user's and other user's checked out items can be download too.

If local working folder have the same name of file, a confirm box appears.

Deleting category and file

Select **[Delete...]** from the right click popup menu on the category or file to delete. The Delete Items dialog shows items to be deleted.

Select items to delete and click **[Delete]** and the selected items or category will be deleted from the repository server.

Changing name

Select **[Rename...]** from the right click popup menu on the category or file to rename. The Rename this category or Rename this item dialog appears. Select items to rename and click **[Confirm]**.

OZ repository doesn't allow concurrent modifications or updates by multiple designers in order to keep consistency of file sharing and updating. Repository server allows only one designer to edit file at any time. So, in order to edit files on the server, the designer must get update permission from the repository server.

Check in

Check in is a process to save the modified file and return the update permission to repository server. Click **[Check In...]** from the right click popup menu on the file to check in.

Check files and click **[Check in]** in Item check in dialog.

When you edited a checked out file, and if you want to save the changes, but not return the permission, check "Keep checked out" option and click **[Check in]**. This option eliminates the hassle of checking in and checking out again.

Check out

Checkout is a process to get the permission to edit and update repository server files from OZ repository according to its rules.

Click **[Check Out...]** from the right click popup menu on the file to check out. Check files and click **[Check out]** in Item check out dialog.

Undo check out

Undo Checkout returns the edit permission without saving the changes of checkout file to repository server.

Click **[Undo Check Out...]** from the right click popup menu on the file to undo check out. Check files and click **[Check out]** in Undo check out item dialog.

Once item got history by checking in and checking out, you can see them using **Show History** menu.

Select **[Show History...]** from the right click popup menu on the item in repository server.

A dialog shows a history of the selected item.

The versions of an item history is given at item check in time and the lesser number, the older version.
You can select a version and download, view comment, rollback, delete history.
You can also compare between scripts of two versions.

Select **[Find Items...]** from the right click popup menu on the category to search in repository server.

Find item dialog appears. Path shows the selected category path. Click **[Find]** after setting up Text, Default option, Item option and selected items are listed.

You can do Check in, Check out, or Cancel Check out by right click popup menu on an item.

Caution for item search

- Version information will be shown only when"Comment" and "Include item history" option were checked. Otherwise not be shown. The lesser version number, the older version. The right click popup menu on an item having history versions have only **[Download...]** and **[Comment...]** menu activated.
- The None type server, since it does not support history functions such as Check in and Check out, Find item dialog shows all options in "Item option" inactivated.
- Item search function can be used only for repository server, not for local working folder.
- "Include item history" option must be checked for comment search.

Explains how to set alias information file (db.properties, datasource.properties), XML file path of file store or HTTP store, and CSV file path when adding store.

Using open dialog

Select a file with Open dialog being opened by clicking the file path selection button.

For example, to set alias information file, click the file path selection button. Open dialog appears.

If the disk drive selected in Open dialog (ex. C:\) is the very one on which OZ designer installed, relative path is used and otherwise, absolute path is used.

Entering path

The file path can be set as absolute, relative path, OZ URL.

Absolute path :	C:\conf\db.properties
Relative path :	./db.properties
OZ URL :	ozp://conf/db.properties

Character sets available for encoding and decoding set of database connection are listed. And alias names for each character set also listed.

Character set	Available alias name
ASCII	us-ascii, ascii, 646, iso646-us, default, ascii7
Big5	big5
Cp037	ibm037, ibm-037, cp037, 037
Cp1006	ibm1006, ibm-1006, cp1006, 1006
Cp1025	ibm1025, ibm-1025, cp1025, 1025
Cp1026	ibm1026, ibm-1026, cp1026, 1026
Cp1097	ibm1097, ibm-1097, cp1097, 1097
Cp1098	ibm1098, ibm-1098, cp1098, 1098
Cp1112	ibm1112, ibm-1112, cp1112, 1112
Cp1122	ibm1122, ibm-1122, cp1122, 1122
Cp1123	ibm1123, ibm-1123, cp1123, 1123
Cp1124	ibm1124, ibm-1124, cp1124, 1124
Cp1250	windows-1250
Cp1251	windows-1251
Cp1252	windows-1252
Cp1253	windows-1253
Cp1254	windows-1254
Cp1255	windows-1255
Cp1256	windows-1256
Cp1257	windows-1257
Cp1258	windows-1258
Cp1381	ibm1381, ibm-1381, cp1381, 1381
Cp1383	ibm1383, ibm-1383, cp1383, 1383

Cp273	ibm273, ibm-273, cp273, 273
Cp277	ibm277, ibm-277, cp277, 277
Cp278	ibm278, ibm-278, cp278, 278
Cp280	ibm280, ibm-280, cp280, 280
Cp284	ibm284, ibm-284, cp284, 284
Cp285	ibm285, ibm-285, cp285, 285
Cp297	ibm297, ibm-297, cp297, 297
Cp33722	Ibm33722, ibm-33722, cp33722, 33722
Cp420	ibm420, ibm-420, cp420, 420
Cp424	ibm424, ibm-424, cp424, 424
Cp437	ibm437, ibm-437, cp437, 437
Cp500	ibm500, ibm-500, cp500, 500
Cp737	ibm737, ibm-737, cp737, 737
Cp775	ibm775, ibm-775, cp775, 775
Cp838	ibm838, ibm-838, cp838, 838
Cp850	ibm850, ibm-850, cp850, 850, cspc850multilingual
Cp852	ibm852, ibm-852, cp852, 852, cspcp852
Cp855	ibm855, ibm-855, cp855, 855, cspcp855
Cp856	ibm856 ibm-856, cp856 856
Cp857	ibm857, ibm-857, cp857, 857, csibm857
Cp860	ibm860, ibm-860, cp860, 860, csibm860
Cp861	ibm861, ibm-861, cp861, cp-is, 861, csibm861
Cp862	ibm862, ibm-862, cp862, cp862, 862, cspc862latinhebrew
Cp863	ibm863, ibm-863, cp863, 863, csibm863
Cp864	ibm864, ibm-864, cp864, csibm864
Cp865	ibm865, ibm-865, cp865, 865, csibm865
Cp866	ibm866, ibm-866, cp866, 866, csibm866
Cp868	ibm868, ibm-868, cp868, 868

Cp869	ibm869, ibm-869, cp869, 869, cp-gr, csibm869
Cp870	ibm870, ibm-870, cp870, 870
Cp871	ibm871, ibm-871, cp871, 871
Cp874	ibm874, ibm-874, cp874, 874
Cp875	ibm875, ibm-875, cp875, 875
Cp918	ibm918, ibm-918, cp918, 918
Cp921	ibm921, ibm-921, cp921, 921
Cp922	ibm922, ibm-922, cp922, 922
Cp930	ibm930, ibm-930, cp930, 930
Cp933	ibm933, ibm-933, cp933, 933
Cp935	ibm935, ibm-935, cp935, 935
Cp937	ibm937, ibm-937, cp937, 937
Cp939	ibm939, ibm-939, cp939, 939
Cp942	ibm942, ibm-942, cp942, 942
Cp942C	cp942c
Cp943	ibm943, ibm-943, cp943, 943
Cp943C	cp943c
Cp948	ibm948, ibm-948, cp948, 948
Cp949	ibm949, ibm-949, cp949, 949
Cp949C	cp949c
Cp950	ibm950, ibm-950, cp950, 950
Cp964	ibm964, ibm-964, cp964, 964
Cp970	ibm970, ibm-970, cp970, 970
EUC_CN	gb2312, gb2312-80, gb2312-1980, euc-cn, euccn
EUC_JP	eucjis, euc-jp, eucjp, extended_unix_code_packed_format_for_japanese, cseucpkdfmtjapanese, x-euc-jp, x-eucjp
EUC_KR	ksc5601, euc-kr, euckr, ks_c_5601-1987, ksc5601-1987, ksc5601_1987, ksc_5601, 5601

EUC_TW	cns11643, euc-tw, euctw
ISO2022JP	jis, iso-2022-jp, csiso2022jp, jis_encoding, csjisencoding
ISO8859_1	8859_1, iso_8859-1:1978, iso-ir-100, iso_8859-1, iso-8859-1, iso8859-1, latin1, l1, ibm819, ibm-819, cp819, 819, csisolatin1
ISO8859_15_FDIS	8859_15, iso-8859-15, iso_8859-15, iso8859-15, ibm923, ibm-923, cp923, 923, latin0, latin9, csisolatin0, csisolatin9
ISO8859_2	8859_2, iso_8859-2:1987, iso-ir-101, iso_8859-2, iso-8859-2, iso8859-2, latin2, l2, ibm912, ibm-912, cp912, 912, csisolatin2
ISO8859_3	8859_3, iso_8859-3:1988, iso-ir-109, iso_8859-3, iso-8859-3, iso8859-3, latin3, l3, ibm913, ibm-913, cp913, 913, csisolatin3
ISO8859_4	8859_4, iso_8859-4:1988, iso-ir-110, iso_8859-4, iso-8859-4, iso8859-4, latin4, l4, ibm914, ibm-914, cp914, 914, csisolatin4
ISO8859_5	8859_5, iso_8859-5:1988, iso-ir-144, iso_8859-5, iso-8859-5,iso8859-5, cyrillic, csisolatincyrillic, ibm915, ibm-915, cp915, 915
ISO8859_6	8859_6, iso_8859-6:1987, iso-ir-127, iso_8859-6, iso-8859-6, iso8859-6, ecma-114, asmo-708, arabic, csisolatinarabic, ibm1089, ibm-1089, cp1089, 1089
ISO8859_7	8859_7, iso_8859-7:1987, iso-ir-126, iso_8859-7, iso-8859-7, iso8859-7, elot_928, ecma-118, greek, greek8, csisolatingreek, ibm813, ibm-813, cp813, 813
ISO8859_8	8859_8, iso_8859-8:1988, iso-ir-138, iso_8859-8, iso-8859-8, iso8859-8, hebrew, csisolatinhebrew, ibm916, ibm-916, cp916, 916
ISO8859_9	8859_9, iso-ir-148, iso_8859-9, iso-8859-9, iso8859-9, latin5, l5, ibm920, ibm-920, cp920, 920, csisolatin5
ISO8859_13	8859_13, iso_8859-13, iso-8859-13, iso8859-13,
JISAutoDetect	jis auto detect
Johab	ksc5601-1992, ksc5601_1992, ms1361

KOI8_R	koi8-r, koi8, cskoi8r
MS874	windows-874
MS932	shift_jis, ms_kanji, csshiftjis, windows-31j, cswindows31j, x-sjis
MS949	windows-949
SJIS	shift_jis, pck
TIS620	tis620.2533
UTF8	utf-8, unicode-1-1-utf-8
UTF16	utf-16
Unicode	unicode
UnicodeBig	unicode-1-1, iso-10646-ucs-2, utf-16be, x-utf-16be
UnicodeLittle	utf-16le, x-utf-16le