Chapter

8

The Report Conversion Utility

With the Report Conversion Utility, you can make changes to several aspects of a report by using report macro functions. For example, if the MIS department changed from SQL Server to Oracle, you would use this report conversion method to modify reports to reflect this change. You can change database connections (both named connections and normal connections), databases, owners, table names, and field names. You can also remove fields from your report that have been deleted from a database. While you can make all these changes manually on a report-by-report basis, the Report Conversion Utility allows all reports in a directory to be converted at one time.

Creating the Report Conversion File

The Report Conversion Utility is a function which automatically converts changed reports from the old form to the new one. You no longer need to use the CONN commands in the macro language to create the macro for this purpose. You do need to create the Report Conversion File. However, once you create this file, you can convert all of the affected files at the same time.

When writing the Report Conversion file, it is important to keep in mind the following points:

- There are no spaces between the parameters.
- There must be a carriage return after each line.
- All the information in the setup strings is case sensitive.
- The file must be of type .rcv.

- All strings must be enclosed in double quotes.
- Each string must contain all parameters, even if there is no value. Empty double quotes (" ") indicate that the field has no value. Square brackets [ ] contain the header stating the type of conversion to be made.

The Report Conversion utitlity will not convert reports that meeting the following conditions:

- The report contains report basic macros that reference database fields where the field name has been changed, or the field has been deleted. The conversion utility will convert the field name changes/deletions in the report, but not in the underlying macros.
- If a field is deleted from a database, and the field is referenced in an SQL derived field. The upser has two options to handle this scenario:
    - Review the log created by the conversion utility to review a list of reports not converted and make the changes manually.
    - When running the conversion utility, select the option "If conversion fails, show Fix Repot dialog". When this option is selected, the conversion utility will step when it encounters a report with an SQL derived field that contains a reference to a deleted field. At that point, you can click "Fix Report" to manually delete the SQL derived field in questions. After deleted the SQL derived field, click "Done" and the conversion utility will continue.
    - Reports created with user entered SQL.

There are two types of connections:

- **Named Connections -** converts the reports created when the named connection is changed.
- **Regular Connections -** converts the reports created when an ODBC or Native connection is changed.

Named Connections requires four parameters as described in Table 8.1.

Table 8.1: Parameters for named connections

	Field	Meaning
1	Unique tag	Name of entry in file.
2	Flag	0 (False)- No conversion intended; reference information only. 1 (True)- Convert the named connection in the .RPT file.
3	Old Name	The original named connection.
4	New Name	The changed name of the named connection.

You create a report using 'Development' as the named connection. Later, you change the named connection to 'Production'. The entry in the .rcv file looks like the following:

```
[NamedConnection]
"NamedConn1", 1,"Development","Production"
```

A regular connection requires twelve parameters as described in Table 8.2.

Table 8.2: Parameters for regular connections

Position	Field	Meaning
1	Unique Tag	Name of entry in file.
2	Flag	0 (False)- Conversion not intended; reference information only. 1 (True)- Convert the connection in the .RPT files.
3	Old Connection Type	The original connection type used to create the report. See ‘File|Connection Info’ while in ReportSmith. This is a numeric value, ex: “3” indicates that the original connection type is Excel.
4	Old Server Name	For ODBC connections this is the old ODBC data source name. For remote connections where the connection dialog asks for a server name, this is the old server name.
5	Old User ID	For remote connections where the connection dialog asks for a user ID, this is the original user ID to log on.
6	Old Database	For local ODBC connections, this is the directory name where the tables reside. For remote connections where the connection dialog asks for a database name, this is the original database name.
7	Old Options	If the connection dialog has an ‘Options’ edit field, this is the original information entered in that field.
8	New Connection Type	See ‘Old Connection’ above for more information.
9	New Server Name	For ODBC connections, this is the new ODBC data source name. For remote connections where the connection dialog asks for a server name, this is the new server name.
10	New User ID	For remote connections where the connection asks for a user ID, this is the new user ID used to log on.
11	New Database	For local ODBC connections, this is the new directory where the table resides. For remote connections where the connection dialog asks for a database name, this is the new database name.
12	New Options	If the connection dialog has an ‘Options’ edit field, this is the new information for that field.

You create a report using an ODBC driver with a data source name of Accounting, user ID of Sysadmin, and a database path of c:\accounting. You later change to a new ODBC data source called Budgeting, with a user ID of BudgetAdmin and a path of c:\budget.

The entry in the .rcv file would look like:

```
[Connections]
"RegularConn1",1,55,"Accounting","Sysadmin", "c:\accounting" " ",55,"Budgeting",
"BudgetAdmin","c:\budgeting,""
```

- To create any of the following conversions, you follow the parameters for Regular Connections.

  There are four types of changes that can occur:

  - **Table**
  - **Field rename**
  - **Field delete**
  - **User SQL substitution**

  When there are table change definitions, the fields have the meanings outlined in Table 8.3.

Table 8.3: Field meanings after table change definitions

Position	Field	Meaning
1	Unique Tag	Name of entry in file.
2	Flag	0 (False)- Conversion not intended; reference information only. 1 (True)- Convert the connection in the .RPT files.
3	Connection tag	The connection tag for the table, defined earlier in this file. This must be a regular connection, not a named connection.
4	Old Database Name	The original database name for the table as shown in the table selection box, not the connection dialog box.
5	Old Owner Name	The original owner name for the table as shown in the table selection dialog box.
6	Old Table Name	The original table name as shown in the table selection dialog box.
7	New Database Name	This is the new database name for the table as shown in the table selection dialog box, not the connection dialog box.
8	New Owner	This is the new owner name for the table as shown in the table selection dialog box.
9	New Table Name	This is the new table name for the table.

The following is an example of a table change:

```
[Tables]
"Table1",1,"RegularConn1","Accounting","SysAdm","Payroll","Budgeting","SysAdm",
"PayrollBudget"
"Table2",1,"RegularConn2","Engineering","LabMan","Vendors","Budgeting","LabMan",
"Suppliers"
```

Where there is a Field name change, the fields have the meanings outlined in Table 8.4.

Table 8.4: Field meanings after field name change

Position	Field	Meaning
1	Connection tag	The original connection tag for this field. Must be a regular connection, not a named connection.
2	Table Tag	The original tag for this field.
3	Old Field Name	The original field name.
4	New Field Name	The changed field name.

The following is an example of a field name change definition:

```
[FieldRenames]
"RegularConn1","Table1","OldFieldName","NewFieldName"
"RegularConn2","Table2","OldFieldName","NewFieldName"
```

When a field is deleted, the fields have the meanings outlined in Table 8.5.

Table 8.5: Field meanings when a field is deleted.

Position	Field	Meaning
1	Connection Tag	The connection name for this field. Must be a regular connection not a named connection.
2	Table Tag	The table name for this field.
3	Field Delete	The name of the field you wish to delete.

The following example demonstrates the deletion of a field:

```
[FieldDeletes]
"RegularConn1","Table1","FieldName"
"RegularConn2","Table2","FieldName"
```

When there is a user-entered SQL substitution, the fields have the meanings outlined in Table 8.6.

Table 8.6: Field meanings for user-entered SQL substitutions

Position	Field	Meaning
1		The old SQL
2		The new SQL

The following is an example of user-entered SQL substitutions:

```
[SQLSubs]
" 'TableName'.'FieldName'","'NewTableName'.NewFieldName'"
" 'OtherTableName'.'OtherFieldName'","NewTableName".'NewFieldName'"
```

Example of a Report Conversion File

The Report Conversion Utility takes the conversion information (stored in the Report Conversion File that you create), and uses the macro language to perform the report conversion. As it opens the report, ReportSmith uses the Report Conversion Utility to convert the report. After successfully opening the report and converting it, ReportSmith immediately saves the report and opens the next report to be converted, continuing the process until all reports are converted.

```
[Connections]
;"orasql",1,7,"demo","sysadm","","",6,"cooper","sa","cssdata"
,""
"cheryl",1,6,"cooper","sa","cssdata","",7,"demo","SYSADM","",
""
[Tables]
;"sqlt",1,"orasql","","SYSADM","PS_ABSENCE_HIST","cssdata","d
bo","PS_ABSENCE_HIST"
"fred",1,"cheryl","cssdata","dbo","PS_ABSENCE_HIST","","SYSAD
M","PS_ABSENCE_HIST"
```

Using the Report Conversion Utility

There is a limitation to the Report Conversion Utility. When reports are converted, containing user-entered SQL and tables and/or fields that also need to be converted, SQL execution fails. ReportSmith has two ways of notifying you of this failure — through the Fix Report dialog box or a generated log file. The type of notification that you receive depends on the selections you make as you use the Report Conversion Utility.

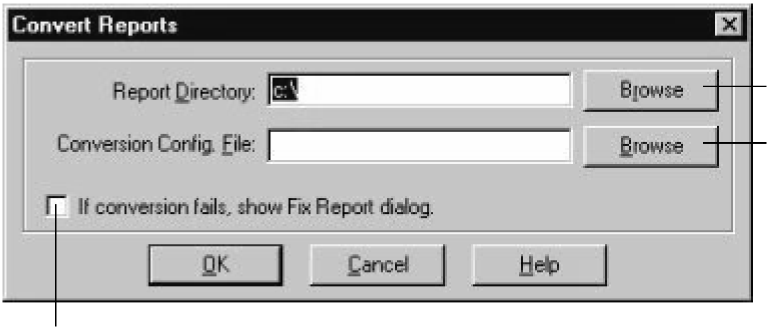
- To use the Report Conversion Utility,

1 Create the Report Conversion File for the report conversion changes you want to make if one has not already been created. Refer to “Creating the Report Conversion File” on page 191 and “Example of a Report Conversion File” on page 197 for help in creating this file.

2 Open the ReportSmit Catalog, and select the CNV button from the toolbar. ReportSmith displays a message box that says “Always backup all .RPT files before converting.”

3 Click OK. ReportSmith displays the Convert Reports dialog box as shown in Figure 8.1.

Figure 8.1: **Convert Reports dialog box**



Convert Reports

Report Directory: c:\

Conversion Config. File:

□ If conversion fails, show Fix Report dialog.

Select location of directory that contains reports to be converted.

Select location of .rcv file (Report Configuration File).

Unchecked - creates error log.
Checked - interrupts report conversion with error message.

- Select the directory that contains the reports you want to convert.
- Select the Conversion Configuration File (the Report Conversion File that you created in step 1).
- Check the box for “If the conversion fails, show Fix Report dialog box.” if you want the option during report conversion of editing a report that could not be converted.

Note: If you check this box and a report conversion fails, ReportSmith immediately displays the Fix Report dialog box with two buttons:

- Fix Report button — Click to edit the report immediately by entering the Query dialog box and modifying the user-entered SQL parts of the report.
- Continue button — Click to continue with the Report Conversion Utility, save the report, and fix the SQL-execution problems at a later time.

If you do not check this box and a report conversion fails, at the end of the report conversion process ReportSmith generates a log file that names all the reports that failed to be converted. You can then edit those reports at a later time.

4 Click OK. ReportSmith closes the Convert Reports dialog box and begins the report conversion process. You see each report in the selected directory opening and closing in succession as each report runs through the conversion process.