



Reporting Guide

Release 2015

Disclaimer

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples are for illustration only and are fictitious. No real association is intended or inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes only.

Sample Code Warranty disclaimer

Microsoft Corporation disclaims any warranty regarding the sample code contained in this documentation, including the warranties of merchantability and fitness for a particular purpose.

License agreement

Use of this software is covered by a license agreement provided with it. If you have any questions, please call the Customer Assistance Department at 800-456-0025 (in the United States or Canada) or +1-701-281-6500.

Copyright

© 2014 Microsoft Corporation. All rights reserved.

Publication Date

September 2014

Contents

Introduction	1
Reporting Overview	1
How to Use this Guide.....	2
Setting up the Default Printer	3
Standard Reporting Options	5
Report Tab Settings	6
Template Tab Settings	8
Sort Tab Settings.....	10
Select Tab Settings	12
Using Wildcards	12
Options Tab Settings.....	13
Cover Page Tab Settings.....	14
Company Selection Tab Settings.....	15
Previewing and Printing Reports	17
Previewing Reports with ROI or ROISRS	17
Printing Reports.....	20
Printing a Report to a File	21
Publishing a Report to a Microsoft SharePoint site.....	23
Maintaining Report Control Records	25
Advanced Reporting Options	29
Microsoft Dynamics SL Reporting Components	29
ROI and ROISRS Software Interaction	30
ROI.exe	30
ROISRS.EXE.....	31
CrpeSol4.dll.....	31
U2LSol4.dll	31
Microsoft.Dynamics.SL.Reporting.dll	32
DynamicsSLFormatting.dll	32
ROI Command Line Parameters	33
Specific Parameters	33
Report File Search Method	36
Date Usage	36
Understanding Pre-Processes	37
What Are Pre-Processes?	37
When Are Pre-Processes Called?	37
Debug Mode.....	37
Pre-process Cancellation of Report Generation.....	37
Database Tables	39
RptControl	39
RptFormat	42
RptRuntime	43
ROIHeader.....	47
ROIDetail	48
Utilizing ROISelect	50
How to Use ROISelect	50
ROISelect Notes	51

Using Crystal Reports	53
Creating a New Report from an Existing One with Crystal Reports	53
Adding a Report to a Menu	55
Create a New Report Control Record	55
Add the New Report to Existing Screens and Reports	56
Add or Modify a Group of Users of the Report	57
Assign Users Access Rights to the New Report	58
Add the Report to a Menu	59
Adding a New Report Format	61
Using SQL Expressions to Expand Reporting Capabilities	62
Hints and Tips	63
Resolving Common Crystal Reports Setup Issues	64
Missing User-Defined Function in Functions Pane	64
Incorrect Logon Parameters Error	64
ODBC Error. Invalid Object Name	64
Text Not Part of Formula Error	65
Purpose of User-Defined Functions	66
How to Use User-Defined Functions	66
Understanding Subreports	67
What Are Subreports?	67
Limits of Subreports	67
Subreport Borders	67
Understanding Drilldown Reports	68
What Are Drilldown Reports?	68
Using Drilldown Reports	68
RIParams	69
Microsoft Dynamics SL RIParams	69
Standard RIParams	71
Creating New RIParams	72
User-defined Functions	74
U2LSol4.dll Functions	74
Crufladg.dll Functions	82
Adding SNote Fields to Crystal Reports	91
Background Information	91
Steps to Add SNote Fields	91
Getting More Crystal Reports Information	92
Using SQL Server Reporting Services	93
ROISRS Features	93
Dynamic Record Selection	93
ROISelect	93
Print Preview	93
Print	94
Export	94
Change a Report Font	94
Print a Report Cover Page	94
Embedding Custom Code	94
Dynamic Sorting	94
Dynamic Report Groupings	94
Dynamic Notes Printing	94
Report Templates	95
Multiple Formats for a Single Report	95
Custom Report Storage	95
Passing Data to a Report	95
Calling Pre-processes and Post-processes	97
Subreport	97
SQL Query as Text	97

Creating a New Report from an Existing One Using SSRS	98
Adding a New SSRS Report to the Menu System.....	100
User-defined Functions for SSRS Reports	101
How User-Defined Functions Are Created	101
How to Use Code-based User-Defined Functions	101
How to Use Assembly-based User-Defined Functions	103
User-Defined Functions Specific to SSRS Reports	105
ROISRS RIParams	111
Microsoft Dynamics SL RIParams.....	111
ROISRS Standard RIParams.....	113
New RIParams.....	114
Screens	115
Printer Options (98.220.00).....	115
Upload to SharePoint Document Library (98.220.01).....	117
Document Library Search.....	117
Document Libraries	118
ROI Screen.....	119
ROI Screen, Report Tab	119
ROI Screen, Template Tab.....	121
ROI Screen, Sort Tab	122
ROI Screen, Select Tab.....	124
ROI Screen, Options Tab.....	127
ROI Screen, Cover Page Tab	128
ROI Screen, Company Selection Tab	129
Field List	131
Report Control Maintenance (98.300.00).....	132
Report Format Tab.....	132
Control Options Tab	133
Custom Fields Tab	135
Appendix A: Converting Custom Reports to Crystal Reports 2008	137
Helpful Tips.....	137
Additional Information	138
Index	139

Introduction

Reporting Overview

One of the outstanding features of Microsoft Dynamics® SL is its powerful financial and business reporting capabilities. Throughout all modules, you will find a comprehensive collection of the reports most useful to managers and decision-makers, all formatted in an easy-to-follow layout that makes finding the information you need fast and efficient.

A wide range of reports is only part of what Microsoft Dynamics SL has to offer. Complementing its many standard reports is an equally broad range of reporting options, all built around a common user interface (UI), which lets you generate reports in practically any format and with as much information as you want. What is more, you decide how you want to view the reports: on screen or on the printed page. You can even send reports to file for later placement and use in programs such as Microsoft® Word or Microsoft® Excel®. Or, you can publish reports to a document library on a Microsoft® SharePoint® site to help your employees gain quick and easy access to the information they need.

What truly sets the software apart is the ability it gives you to customize standard reports—or to add completely new reports—to meet even the most specific reporting needs. Whether you are modifying a standard report for a unique business situation or creating reports for a new supporting application, the software provides the development tools you need to produce useful, professional business and financial reports.

In addition, the ROISRS.exe application processes reports using Microsoft® SQL Server® Reporting Services (SSRS). The ROISRS has the same UI and features as the standard Report Option Interpreter (ROI).

How to Use this Guide

To help you make the most efficient use of this resource, topics are divided into the following areas:

- “Setting up the Default Printer” – Provides step-by-step instructions for setting up the printer you will use to print reports.
- “Standard Reporting Options” – Introduces the reports UI and provides step-by-step instructions for using the many standard reporting options. If you are new to Microsoft Dynamics SL reporting, or you want to learn how to make better use of the standard reporting capabilities, this information is for you.
- “Previewing and Printing Reports” – Provides instructions and other information about how to view and then print a report, either to paper, file, or a Microsoft® SharePoint® site.
- “Maintaining Report Control Records” – Explains how to create and manage the information that determines which options and formats are used when a report is generated.
- “Advanced Reporting Options” – Describes the purpose and functionality of each software component in the reporting system and how they interact during the reporting process. If you are a developer, this information is an excellent reference.
- “Using Crystal Reports” – Explains key concepts specific to Microsoft Dynamics SL’s report delivery engine, the industry-standard Crystal Reports system. It also includes many different Crystal Reports reference information that will be invaluable to you as you use Crystal Reports to modify reports or develop new ones.
- “Using SQL Server Reporting Services” – Provides information about the SQL Server Reporting Services (SSRS) report delivery engine. It includes information that will aid you in developing new SSRS reports or modify existing reports.
- “Screens” – Describes screens that are used in the report generation process.

Setting up the Default Printer

Before you print reports, forms, or labels, you must first define the default printer settings for your whole Microsoft Dynamics SL system. These settings include the default output destination, printing font, and page orientation. Remember that the default Windows printer and the default Microsoft Dynamics SL printer can be one in the same. Also, the software provides the option to designate separate printers for each.

Note: When you print a single report, you can override default printer settings.

To set up the default Microsoft Dynamics SL printer:

1. Choose **Application | Printer Setup** from the toolbar. *Printer Options (98.220.00)* appears.

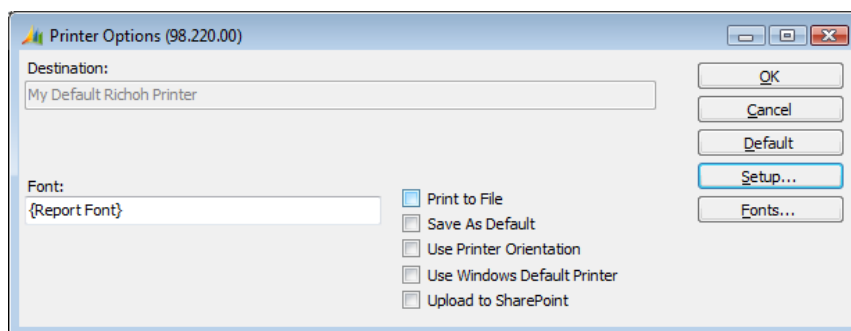


Figure 1: Printer Options (98.220.00)

2. Click **Setup**. The *Print Setup* dialog box appears.

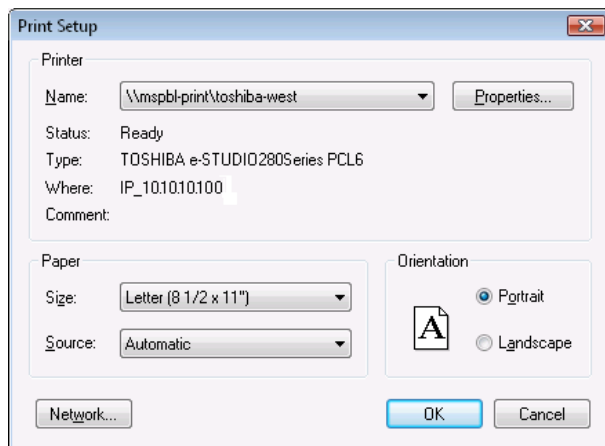


Figure 2: Print Setup dialog box

3. Select a printer to use as the default printer. If you do not see a list of printers, or if you cannot connect to the printer you want to use, contact your system administrator.
4. Select the paper size, source, and orientation.
5. Click **Properties**, if available, and define any advanced printer properties.
6. Click **OK** on each dialog that you use to return to *Printer Options (98.220.00)*.

7. Click **Fonts**. The *Font* dialog appears.

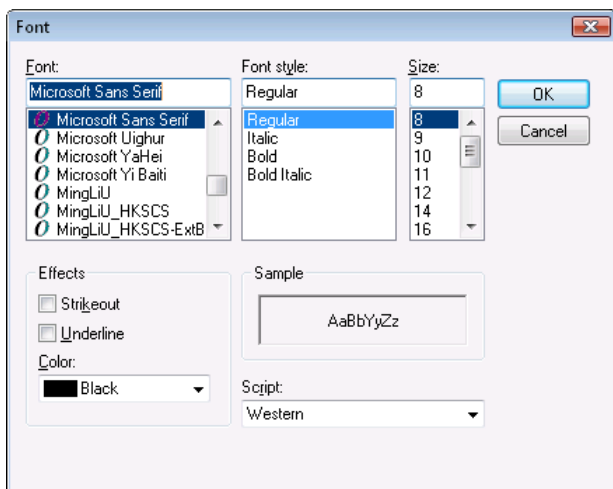


Figure 3: Font dialog

1. Define the default report font, font style, and font size. Click **OK** to accept your settings and return to *Printer Options* (98.220.00).
2. Select Save As Default.
3. Click **Ok** to close *Printer Options* (98.220.00).

Standard Reporting Options

The Report Option Interpreter (ROI) is the UI to Microsoft Dynamics SL's powerful reporting capabilities. It provides a consistent, easily navigated reporting "front end" that lets you print standard reports efficiently. It also enables you to modify the content and format of the standard reports, using the many reporting options, to meet specific reporting requirements. It features the following helpful user-controlled reporting capabilities:

- Sort a report by any existing report field
- Select records for printing based on any report field
- Print character masks for certain special form-type reports
- Include notes on reports
- Include all data items on reports, ignoring Quick Send preferences defined for customers, employees, projects, and vendors
- Include all data items on reports, ignoring Doc Share settings defined for customers, projects, and vendors
- Select report data based on period information
- Select report data based on a date or range of dates
- Save current ROI window settings in a report template and retrieve them later for printing a new edition of the same report according to the same settings
- Print a report cover page that describes the report's content, content sorting criteria, and content selection criteria
- Limit the number of report pages printed: a single page or a range of pages
- Print individual or combined reports for companies you select
- Print multiple copies of the same report
- Print a report to any available printer
- Print a report to a file: ASCII text or several other file formats
- Publish a report to a Microsoft® SharePoint® site

The standard ROI UI consists of seven tabs that appear in the report window:

- Template
- Report
- Sort
- Select
- Options
- Cover Page
- Company Selection

The **Template**, **Report**, **Sort**, **Select**, and **Cover Page** tabs are always active, regardless of which reporting window you are using. The **Options** and **Company Selection** tabs have a more specific purpose in Microsoft Dynamics SL reporting. Whether they are active depends on the reporting window you have open.

Each of these ROI tabs is explained in the topics that follow.

Note: For each report window, the ROI interface is set to display the **Report tab** in front of all other ROI tabs when you open the window, unless you have previously defined a report template for the window. In such a case, the report window always opens with the **Template tab** on top (see "**Template Tab Settings**" for more information).

Report Tab Settings

Use the **Report** tab to define options that control the appearance and content of the standard report, for example:

- Print report information in summary or detail on plain paper or pre-printed forms.
- Print notes that consist of non-financial, user-defined information attached to data records in windows (as in a supervisor note that is attached to a customer ID and states that the customer is an excellent credit risk).
- Print a specific report date in the report's header block.
- Print a specific report page or page range.
- Print a report based on a period or range of periods.
- Print a report based on a date or range of dates.

Also use the **Report** tab to control how many copies of the report are produced after you start the printing process.

For some reports, the **Report** tab contains options that let you print the report for a specific range of fiscal periods (for example, periods 03 through 06), for a specific fiscal period only, or for a specific fiscal year. For form-related printing (checks, for example), it includes an option that lets you print a dummy character mask to verify form alignment in the printer before you print the actual forms. For reports that can be sent electronically through Quick Send or published to Microsoft® SharePoint® sites by using Doc Share, it includes options to ignore Quick Send preferences and Doc Share settings so that all data items are printed on the report.

For more information about this tab, see “ROI Screen, Report Tab” on page 119.

To define report settings:

1. On the Microsoft Dynamics SL window, open a module group, and then select a module.
2. On the application pane, locate **Reports**, which lists the module's available standard reports.
3. Choose the name of the report to print (for example, **Sales by Commission**). This opens the *Report Interface* at the **Report** tab.

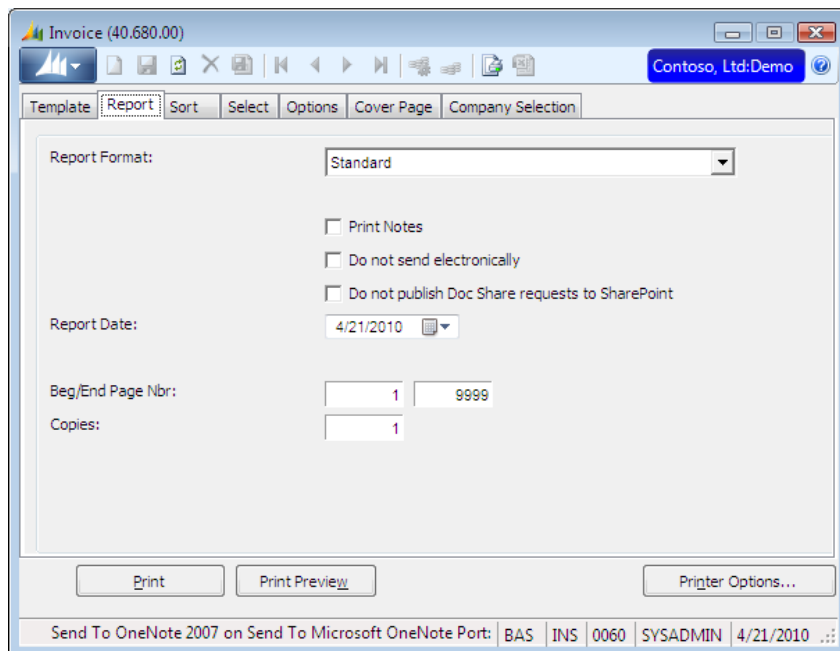


Figure 4: Report tab

4. Select a format from the **Report Format** list.

5. Select **Print Notes** to print notes attached to data items. This check box is not available if there are no **Notes** fields on the specified report.
6. Select **Do not send electronically** to include all data items on the report. Quick Send preferences defined for customers, employees, projects, and vendors are ignored. This check box is not available if document types are not defined on *Quick Send Setup* (21.951.00) in the Shared Information module.
7. Select **Do not publish Doc Share requests to SharePoint** to include all data items on the report. Doc Share settings defined for customers, projects, and vendors are ignored. This check box is not available if entity types are not defined on *SharePoint Site Configuration* (98.360.00).
8. Type the date to appear in the report header in **Report Date**. The current date is the default. To create a date formula, press F2. *Relative Date* (98.240.00) displays.
 - Select **Relative** or **Absolute** from the **Day**, **Month**, and **Year** lists.
 - Click the **Up** or **Down** arrows to the right of **Day**, **Month**, and **Year** to select the relative or absolute values.
 - Under **Example**, **Current** shows the current date and **Result** shows the resulting date based on the numbers you have selected in **Day**, **Month**, and **Year**.
9. Type the beginning and ending fiscal periods to report in **Beg/End Period**, if included. To create a period formula, press F2. *Relative Period* (98.250.00) appears.
 - Select **Relative** or **Absolute** from the **Month** and **Year** lists.
 - Click the **Up** or **Down** arrows to the right of **Month** and **Year** to select the relative or absolute values.
 - Under **Example**, **Current** shows the current month and year and **Result** shows the resulting month and year based on the numbers you have selected in **Month** and **Year**.
10. Type the period to report in **Period to Report**, if included.
11. Type the fiscal year to report in **Fiscal Year**, if included.
12. Type the page range to print in **Beg/End Page Nbr**.
13. Type the number of copies to print in **Copies**.
14. If the **Report** tab has any other report-specific options (for example, **Print Mask** for printing character alignment masks on forms), define the appropriate settings for these.

Template Tab Settings

Use the **Template** tab to define a template of the report's current print settings (for example, the options selected in all other ROI tabs) in order to speed up printing the report in the future. Defining a report template saves the current print settings in the database under a unique template ID and description. If you ever want to reprint the report using these same settings, select the template ID and ROI will print the report again exactly as it is currently defined.

Report templates save time (you do not have to redefine all the print settings) and reduce printing errors (for example, printing the report using the wrong format or for the wrong range of fiscal periods). They also save time when you define additional report templates (for example, you define a new report template that uses most of the same print settings as an existing report template. However,, the new template sends its output to a different printer than the one used by the original template).

For more information about this tab, see "ROI Screen, Template Tab" on page 121.

To save a report template:

1. Click the **Template** tab of the selected report.

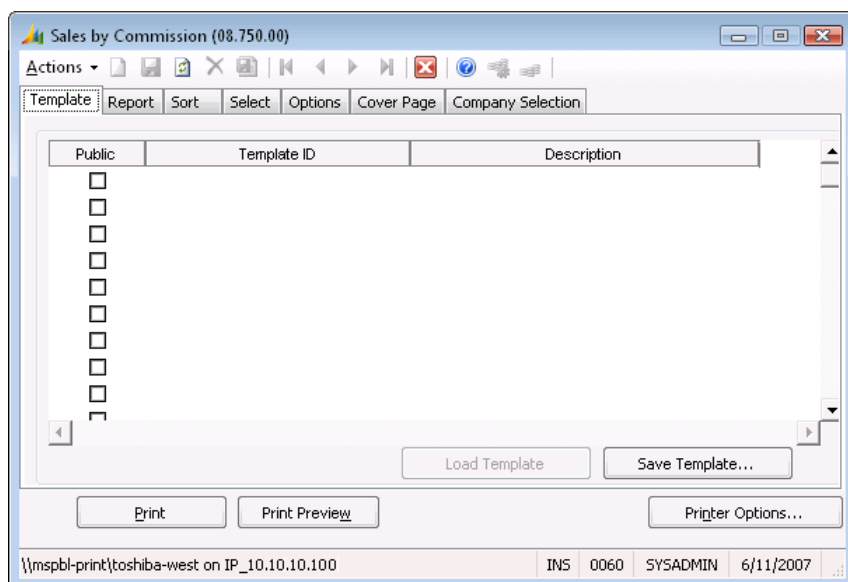


Figure 5: Template tab

2. On the **Template** tab, click **Save Template**. *Save Template (98.600.01)* appears.

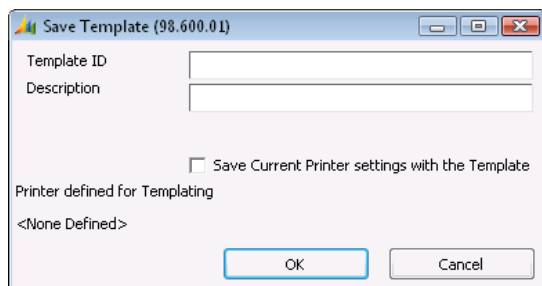


Figure 6: Save Template (98.600.01)

3. Type a template ID in **Template ID**.
4. Type a description of the template in **Description**.
5. Select **Public** to make the template available to all users. Otherwise the template is available only to the user who created it.
6. Select **Save Current Printer settings with the Template** to let the template use the printer that is set up in *Printer Options (98.220.00)*.

Note: It is best not to save the settings with the template if this is a public report and not all users can connect to the printer.

7. Click **OK**.

To use a report template:

1. Click the **Template** tab.

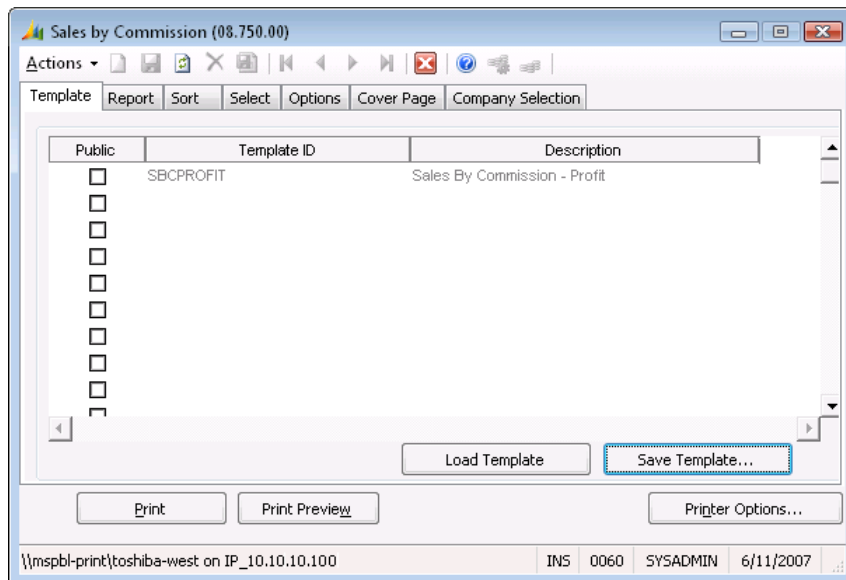


Figure 7: Selecting a report template from the Template tab

2. Highlight the template ID to use.
3. Click Load Template.
4. Make any changes to the report, if necessary.
5. Click **Print**.

Note: The **Print** button displays as **Upload** when **Upload to SharePoint** is selected in *Printer Options* (98.220.00).

Sort Tab Settings

Use the **Sort** tab to define a custom sort order for report information based on any of the report's record.filename fields. For example, the standard sort order priority for customer names is last name first, first name second. With the **Sort** tab, you could set up a report to sort customers based on the sort order first name first. Or you might set up a report to sort customers based on year-to-date net sales.

The **Sort** tab lets you define custom report sort orders based on multiple record.filename fields. For example, first sort a report alphabetically based on vendor name, then sort the report based on year-to-date net purchases, with the lowest net purchase amount listed first.

The **Sort** tab is initialized to display grouping and sorting criteria as defined in the report itself. This lets you make modifications, additions, or deletions to the criteria and get the complete picture of how the report will be grouped and sorted.

Example: By default, the software sorts the *Customer Trial Balance (08.620.00)* report in the Accounts Receivable module first on subaccount number, second on customer ID, third on reference number, and fourth on transaction date. To sort the report first by customer ID, select CustID as the first field to sort.

Note: The available custom sort options depend largely on the structure of the report. Because of this, a report you create by using custom sort options may not appear as you intend. To make sure that a custom-sorted report will print as you expect, preview the report using the custom sort options before sending the report to the printer.

For more information about this tab, see “ROI Screen, Sort Tab” on page 122.

To specify sort order:

1. Click the **Sort** tab.

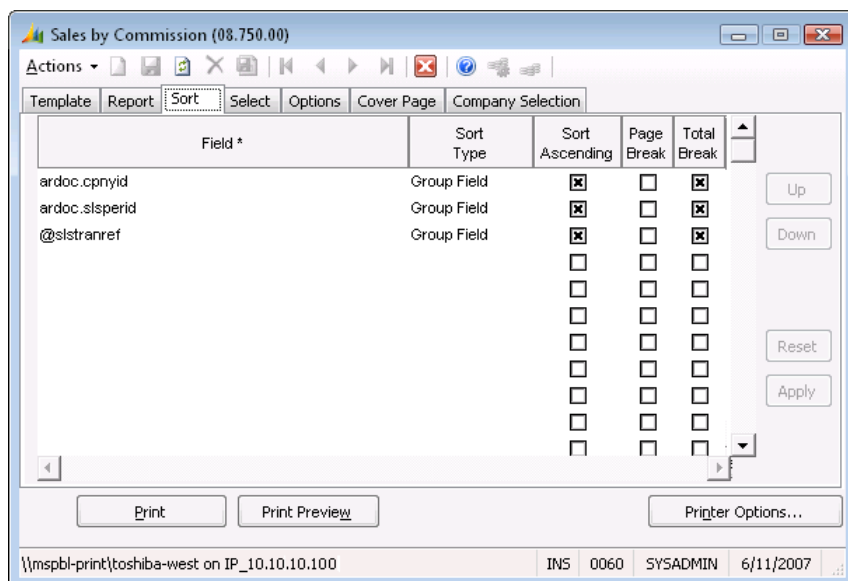


Figure 8: Sort tab

2. Type a field name in Field or press F3 to open a Possible Values (PV) window listing all record.fieldnames for the report. Select a record.fieldname, and then click **OK**.
3. Select Group Field or Sort Field in **Sort Type**.
4. Select **Sort Ascending** to sort a column of data in ascending order.
5. Select **Page Break** to insert a page break when the value in a field changes.
6. Select **Total Break** to insert a total report break when the value in the field changes. For a new group, this makes the group footer section visible, but nothing is added to the section.

7. Select a field name and use the **Up** and **Down** buttons to move its row up or down.
8. If you have made changes that you do not want to apply, click **Reset**. This restores grouping and sorting criteria from the report.
9. To apply your changes for this specific report generation, click **Apply**.

Select Tab Settings

Use the **Select** tab to print a report that contains only a subset of the report's available information (for example, only a portion of the total information maintained for that report in the database). For example, instead of printing all vendor information, you might want to print a *Vendors* (03.670.00) report listing only those vendors located in Ohio. Or you might want to print a *Customer* (08.650.00) report listing only customers who have year-to-date sales of \$50,000 or more.

On the **Select** tab, you identify the information to be reported by defining one or more query-like restriction clauses that qualify as ROI search criteria (for example, Sales.SAL Is greater than \$1,000). For each report, you can select information the information to report based on any of the report's record.filename fields.

For more information about this tab, see "ROI Screen, Select Tab" on page 124.

To select values:

1. Click the **Select** tab.

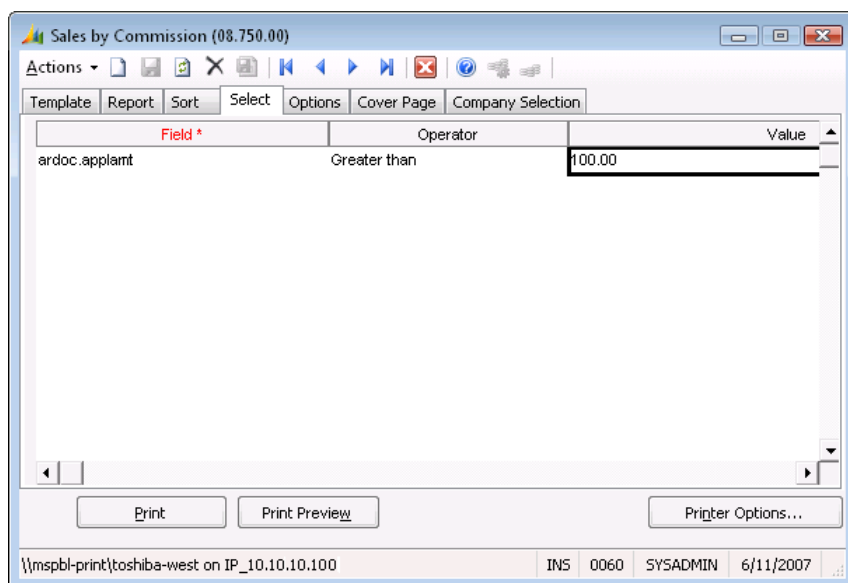


Figure 9: Select tab

2. Type a field name in **Field** or press F3 to open a Possible Values (PV) window listing all record.fieldnames for the report. Select a record.fieldname, and then click **OK**.
3. Select a standard operator from the **Operator** list.
4. Type the value or range of values. For a range of values, separate the values with a comma.
5. If defining a range, select either AND or OR from the **Boolean** list.
6. If present, click **View Selections** to check the selection criteria.

Using Wildcards

Use wildcards in the **Value** field to select ranges of values. Wildcards include the asterisk (*) and the question mark (?).

Example: To select all account numbers with a 1 in the first character position and a 2 in the third character position use the wildcards 1*2*, 1?2? or 1*2? in the **Value** field.

Options Tab Settings

Use the **Options** tab to select report-specific additional report options that further control the content of reports. For example, you might use options on the **Options** tab to exclude tax information, age documents by invoice date, or include trade discount amounts.

The options appearing on the **Options** tab vary from report to report, and not all reports have additional report options. If a report has additional report options, the **Options** tab will be available. Otherwise, this tab will be unavailable.

Note: If you are a developer, you can use the **Options** tab to modify the additional report options of the standard reports or add as many as five additional report options to the custom reports you develop. You can also use the **Options** tab to make a report's information available to other processes. See "Advanced Reporting Options" for more information.

The fields on the **Options** tab vary, depending on the purpose of the report. For an explanation of each field on the **Options** tab, see the appropriate module's Help or user's guide.

For more information about this tab, see "ROI Screen, Options Tab" on page 127.

To set options:

1. Click the **Options** tab.

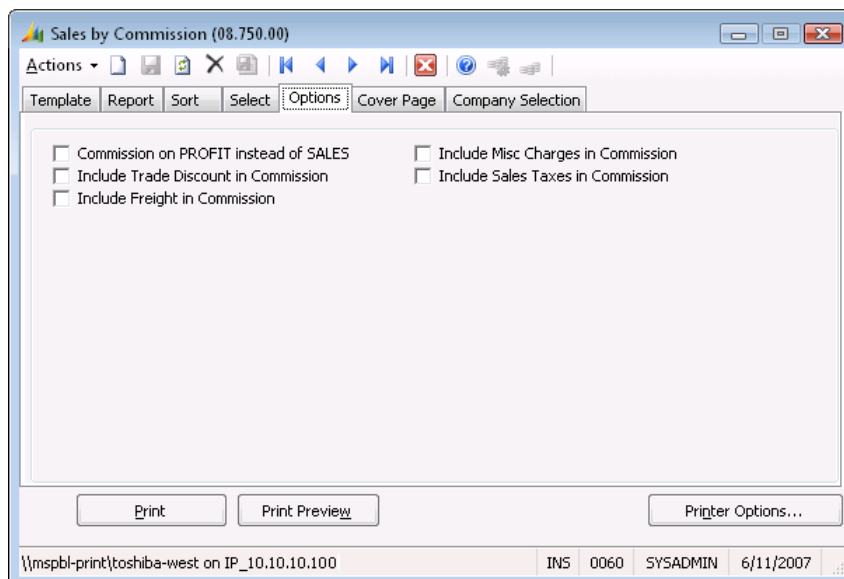


Figure 10: Options tab

2. Select the options you want to use.

Cover Page Tab Settings

Use the **Cover Page** tab to print a cover page for the current report. A report cover page is an introduction to the report, providing a summary of the report's content, selection criteria, and sort order. It also contains any user-defined comments defined in the **Cover Page** tab prior to printing the report. The report 98990.rpt or 98991.rdl prints if you select **Print Cover Page**.

For more information about this tab, see "ROI Screen, Cover Page Tab" on page 128.

To print a cover page:

1. Click the **Cover Page** tab.

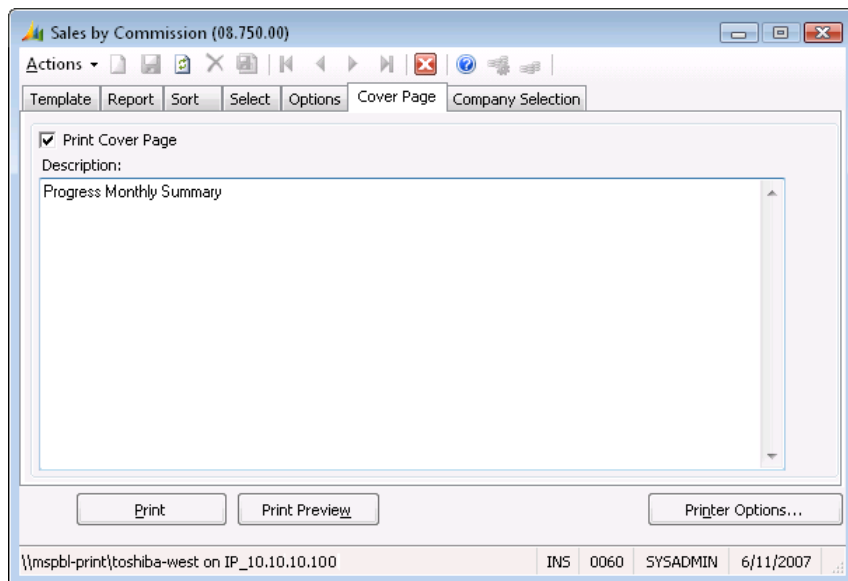


Figure 11: Cover Page tab

2. Select **Print Cover Page** to print a cover page.
3. Type the text to print on the cover page in the **Description** text box. The text prints on the cover page in addition to the sort and selection criteria.

Company Selection Tab Settings

The **Company Selection** tab is available if **Multi-Company Selection Allowed** is selected on *Report Control Maintenance* (98.300.00), **Control Options** tab. Use the **Company Selection** tab to choose options for company-based reporting, either on a multi-company report or on a single report for each company you select. On this tab, you can list all of the companies that share the application database you are currently using. Alternatively, select **Report per Company** to choose from a list of all companies that share the current system database. You can select one, several, or all of the companies for reporting.

For more information about this tab, see “ROI Screen, Company Selection Tab” on page 129.

To select a company or companies for reporting:

1. Click the **Company Selection** tab. **Current Company** is selected by default.

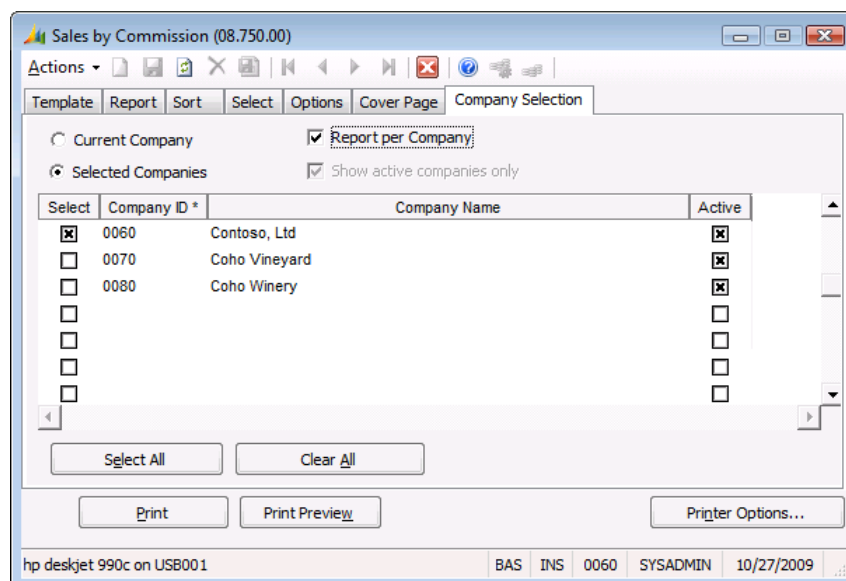


Figure 12: Company Selection tab

2. To select companies for a combined multi-company report, click **Selected Companies**.
– OR –
To select companies for individual reports, select **Report per Company**.
3. Select from the grid the companies on which you want to report.

Previewing and Printing Reports

After defining a report, you can preview it to verify that it will give you the information you need before printing it to a printer, a file, or a document library on a Microsoft® SharePoint® site. You can set a default Report Format by using *Default Report Format Maintenance* (21.400.00). For more information about how to set a default report format see “Setting up Default Report Formats” in Shared Information Help or user’s guide.

By default, the software prints the report based on the default printer setup. To choose a different printer orientation or font to fit the report on the page, override the default printer setup. For more information about printer setup, see “Setting up the Default Printer” on page 3.

Note: Printing defaults exist in the [Print Default] section of the Solomon.ini file. For more information about the Solomon.ini file, see “Appendix A: Solomon.ini Settings” in the System Manager Help or user’s guide.

Previewing Reports with ROI or ROISRS

Before you print a report, you can preview its format and content on screen in the *Print Preview* window. This enables you to save time as well as paper. Continue previewing a report until you get it exactly the way you want it, then send it to the printer for a report that is right the first time you print it. To preview a report, click **Print Preview** in the report window.

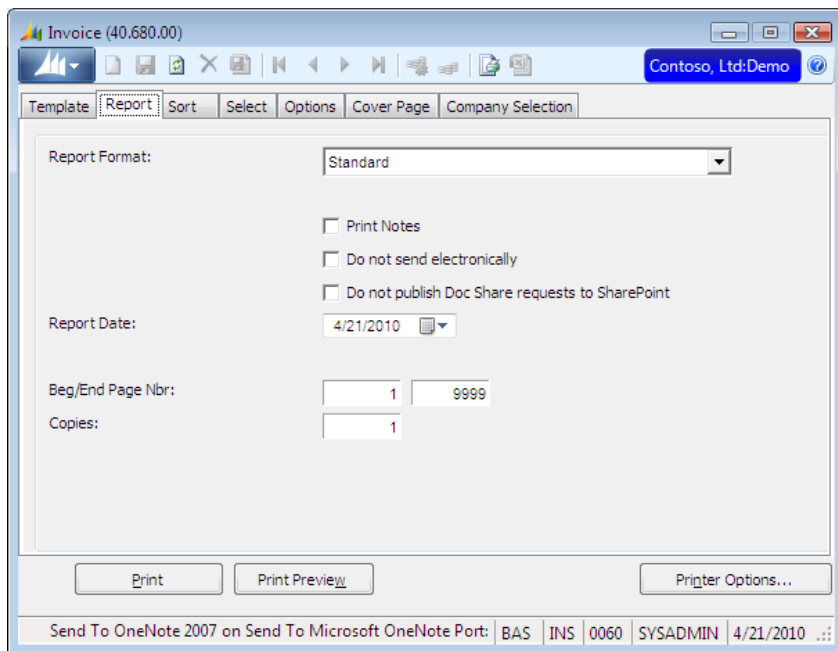


Figure 13: Print Preview button on the Report tab

The report processes according to the settings you select on the **Template** through **Company Selection** tabs. Then the *Print Preview* window opens, showing you the report with the format and content that is used when you send the report to the printer.

Account	Subaccount	Description	Beginning Balance	Period Activity	Ending Balance	Adjustment	Adjustment Balance
				Debit	Credit		
1030	00-000-00-00-0	Cash-US Dollars-US	12,345,678.90	0.00	0.00		12,345,678.90
1030	01-100-AA-00-00-1	Cash-US Dollars-US	-2,221,396.15	515,757.88	154,219.97		-1,859,856.44
1030	03-440-AA-00-51-0	Cash-US Dollars-US	0.00	6,000,000.00	0.00		6,000,000.00
1031	01-100-AA-00-00-1	Cash US Dollars	3,250,000.00	0.00	0.00		3,250,000.00
1031	03-440-AA-00-51-0	Cash US Dollars	11,828,000.00	5,469,550.00	0.00		17,097,550.00
1037	00-000-00-00-000	Cash US Dollars-Main	625,321.94	0.00	0.00		625,321.94

Figure 14: Report preview before printing - ROI

The name of the report appears in the window title bar. Underneath the title bar is a row of buttons that you can use to navigate and manage the report preview.














Button		Function
Export Report		Export the contents of the report, including formatting, to a file in several different file formats, such as a Word .doc file or an Adobe .pdf file.
Email Report		Send a report that is exported to a file with the contents of the report, including formatting, in several different file formats.
Print Report		Print all pages of the report you are viewing.
Refresh		Obtain new data for your report.
Toggle Parameter Panel		Open or close an area that lists formatting and filtering parameters associated with the report.
Toggle Group Tree		Open or close the group tree.
Go to First Page		Display the first page of the report you are viewing.
Go to Previous Page		Display the previous page of the report you are viewing.
Go to Next Page		Display the next page of the report you are viewing.
Go to Last Page		Display the last page of the report you are viewing.
Go to Page	<input type="text" value="1"/> /2	Select a page to display.
Find Text		Search for specific text in the report you are viewing.
Zoom		Increase or decrease the display size of the report you are viewing (for example, select 200% to magnify the report to two times its normal viewing size).

SSRS Report Viewer

Account	Subaccount	Description	Beginning Balance	Period Activity	Ending Balance	Adjustment	Adjustment Balance
				Debit	Credit		
1000	00-000-00-00-00-0	Cash-US Dollars-US	12,345,678.90	0.00	0.00		12,345,678.90
1000	01-500-AA-00-00-1	Cash-US Dollars-US	-2,201,395.15	381,537.71	0.00		-1,819,857.44
1000	03-440-AA-00-51-0	Cash-US Dollars-US	0.00	6,000,000.00	0.00		6,000,000.00
1001	01-500-AA-00-00-1	Cash US Dollars	3,250,000.00	0.00	0.00		3,250,000.00
1001	03-440-AA-00-51-0	Cash US Dollars	11,628,000.00	6,488,550.00	0.00		17,097,550.00
1007	00-000-00-00-00-0	Cash US Dollars-Main	625,321.94	0.00	0.00		625,321.94
1007	01-500-AA-00-00-1	Cash US Dollars-Main	-25,341.85	0.00	0.00		-25,341.85

Figure 15: Report preview before printing – SSRS Report Viewer

Use the SSRS Viewer toolbar to navigate and manage the report preview.

Button	Function
Show or Hide Document Map	 Opens and closes the group tree
First Page	 Displays the first page of the report in the Viewer window
Previous Page	 Displays the previous page of the report in the Viewer window
Current Page	<input type="text" value="1"/> of 10 Displays the number of the page you are viewing, as well as the total number of report pages
Next Page	 Displays the next page of the report in the Viewer window
Last Page	 Displays the last page of the report in the Viewer window
Back to Parent Report	 Returns the view from a subreport to the parent report; active only for custom reports that include subreports.
Stop Rendering	 Cancels the data gathering process for the report.
Refresh	 Updates the report you are viewing.
Print	 Prints all pages of the report being previewed
Print Layout	 Toggles the Viewer window to and from print layout.
Page Setup	 Opens Page Setup dialog box for changing page settings
Export	 Export the contents of the report, including formatting, to a file in several different file formats.
Email	 Send a report that is exported to a file with the contents of the report, including formatting, in several different file formats.
Zoom	<input type="text" value="100%"/> Increases or decreases the display size of the report in the Viewer window
Search	<input type="text" value="EMP0133"/> <input type="button" value="Find"/> Lets you search report text for words or phrases

When you are finished with the report preview, click **Close Window** to return to the report window. There you can define new print settings or click **Print** to print the report.

Printing Reports

After you define the report, print it from any tab by selecting to route the report to a printer, a file, or a document library on a Microsoft® SharePoint® site. By default, the software prints the report based on the default printer setup. Preview the report before printing to verify that it will print as you want. To choose a different printer orientation or font to fit the report on the page, override the default printer setup.

To override the default printer setup:

Click **Printer Options**. Change the default printer setup or choose a different printer.

To print the current report:

Click **Print**.

Note: The **Print** button is displayed as **Upload** when **Upload to SharePoint** is selected in *Printer Options* (98.220.00). Reports published to SharePoint sites include all data items, ignoring Quick Send preferences and Doc Share settings defined for customers, employees, projects, and vendors. However, reports sent to printers do not include data items set up for Quick Send or Doc Share. To include these data items on reports, select the **Do not send electronically** or **Do not publish Doc Share requests to SharePoint** check boxes on the **Report** tab. For more information about setting up Quick Send, see “Setting up Quick Send” in the Shared Information online Help or user’s guide. For more information about setting up Doc Share, see “Sharing Documents Using Doc Share” in the System Manager online Help or user’s guide.

Printing a Report to a File

In addition to sending a report to a printer, you can print the report to a text-editable file. You might do this if you want to insert the report into another application such as a Microsoft Word document or an Excel spreadsheet.

To print the current report to a file:

1. Click **Printer Options**. *Printer Options (98.220.00)* appears.

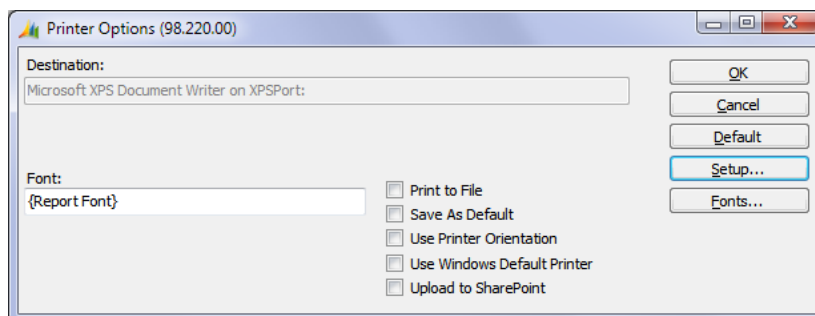


Figure 16: Typical Printer Options (98.220.00)

2. Select **Print to File**, and then click **Setup**. The *Print to File* window appears.

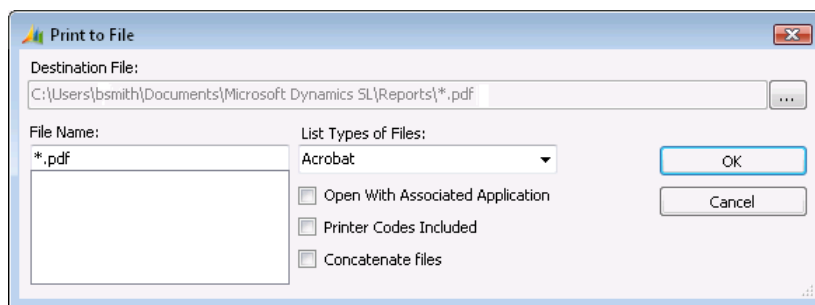


Figure 17: Print to File window

3. Define the print file name at **File Name**.
4. Use **Directories** and **Drives** to identify where you want to create the report's print file.
5. In **List Types of Files**, choose the file type you want for the report file.

Note: The supported file formats for SSRS reports are PDF and XLS.

6. If you want the report to be opened automatically after it is saved to file, click **Open With Associated Application**.
7. Finally, select any additional print file options for the report:
 - **Printer Codes Included** – Sends the printer control codes generated by the report writer to the report file along with the report data.
 - **Concatenate files** – Adds the contents of the current report to an existing print file (for example, a file containing a report produced earlier using the print-to-file function). Concatenation is available only for text files, such as Microsoft Word documents, and not for other file types, such as Excel spreadsheets or Adobe PDF files.
8. Click **OK** until you return to the report's ROI window. Then click **Print** to print the report to the designated file.

Note: Reports printed to files do not include data items set up for Quick Send or Doc Share. To include these data items on reports, select the **Do not send electronically** or **Do not publish Doc Share requests to SharePoint** check boxes on the **Report** tab. For more information about setting up Quick Send, see “Setting up Quick Send” in the Shared Information online Help or user’s guide. For more information about setting up Doc Share, see “Sharing Documents using Doc Share” in the System Manager online Help or user’s guide.

Publishing a Report to a Microsoft SharePoint site

In addition to sending a report to a printer or a file, you can publish the report to a document library on a Microsoft® SharePoint® site. You might do this to help your employees gain quick and easy access to the information they need.

To publish a report to a SharePoint document library:

1. Choose **Application | Printer Setup**. *Printer Options (98.220.00)* appears.

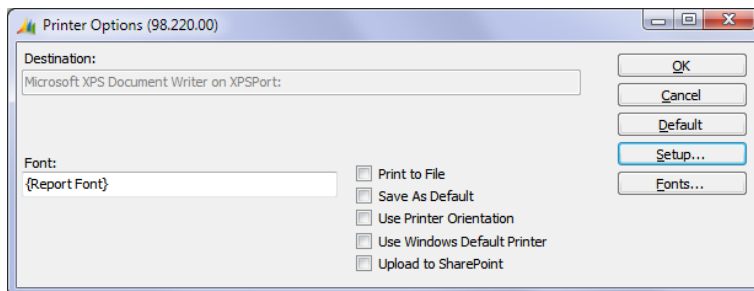


Figure 18: Printer Options (98.220.00)

2. Select **Upload to SharePoint** and then click **Setup**. *Upload to SharePoint Document Library (98.220.01)* appears.

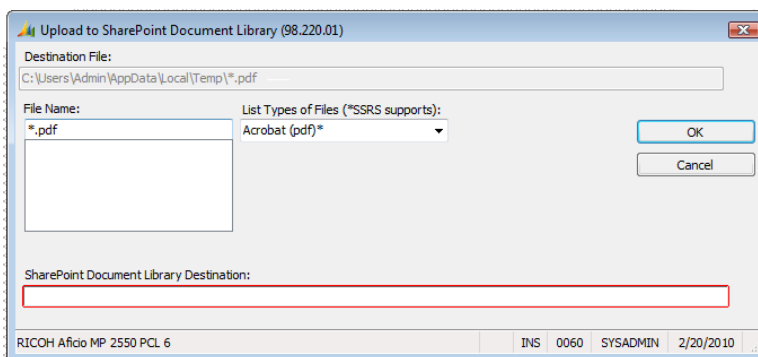


Figure 19: Upload to SharePoint Document Library (98.220.01)

3. In **List Types of Files**, select the type of file to be uploaded to the SharePoint document library.
4. In **SharePoint Document Library Destination**, specify the SharePoint document library path or press **F3** to view a list of previously accessed SharePoint document libraries. *Document Library Search* appears.

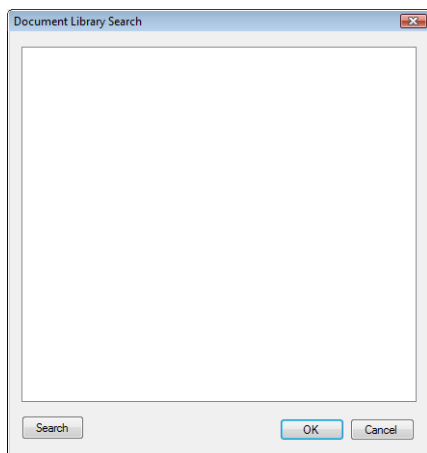


Figure 20: Document Library Search

5. Click **Search** to look up document libraries on a specified SharePoint site. *Document Libraries* appears.

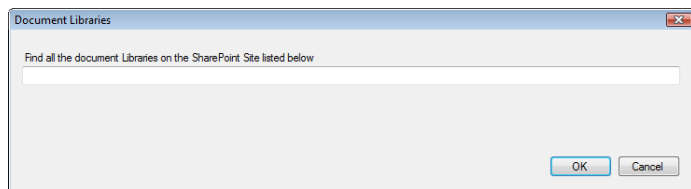


Figure 21: Document Libraries

6. Enter the URL to an existing SharePoint site in **Find all the document Libraries on the SharePoint Site listed below**.
7. Click **OK**. *Document Library Search* appears listing the document libraries on the SharePoint site.
8. Select the appropriate document library and then click **OK**. *Upload to SharePoint Document Library (98.220.01)* appears with the path of the document library in **SharePoint Document Library Destination**.
9. Click **OK** until you return to the report's ROI window. Then click **Upload** to publish the report to the designated SharePoint document library.

Note: Reports published to document libraries on Microsoft® SharePoint® sites include all data items, ignoring Quick Send preferences and Doc Share settings defined for customers, employees, projects, and vendors.

Maintaining Report Control Records

When you print a Microsoft Dynamics SL report from the menu or from within code, a surrogate program, the Report Option Interpreter (ROI), is called. This program starts a printing engine (either Crystal Reports or SQL Server Reporting Services) to print the report. The ROI provides a common interface for all reports and gives you the flexibility to make many choices that affect the appearance of your reports.

Each report can have a different set of options and formats. A record from the RptControl system table determines which options and formats are used when a report is generated. There must be a record in this table for each report printed from Microsoft Dynamics SL.

For more information, see “Report Control Maintenance (98.300.00)” on page 132.

To maintain report control records in *Report Control Maintenance (98.300.00)*:

1. Under **Maintenance** on the System Manager application pane, click **Report Control Maintenance**. *Report Control Maintenance (98.300.00)* appears.

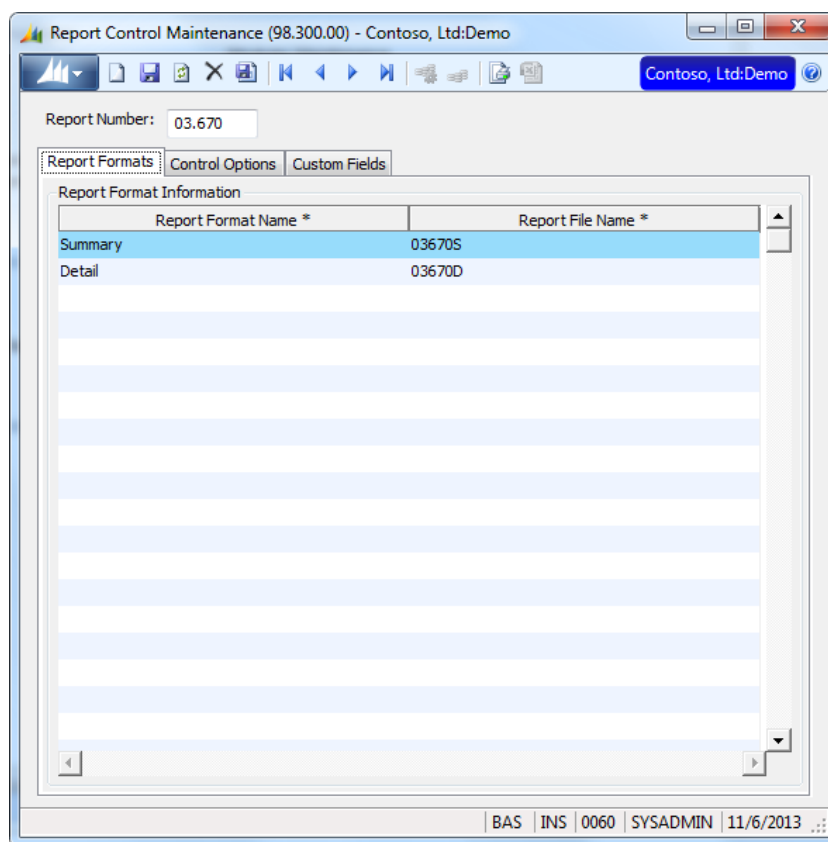


Figure 22: Report Control Maintenance (98.300.00)

2. If you are creating a new report, type a report number in **Report Number**. To select an existing report, Press F3 to open a *Possible Values (PV)* window listing all available report numbers to choose from. Select a report number and then click **OK**.
3. In **Report Format Name**, type a name for the first report format. You may also change the name if this is an existing report.
4. In **Report Format**, type the name of the Crystal Reports definition file for the first report format. This is the report file name without the .rpt extension, or in the case of a SSRS report, the report file name without the .rdl extension.

- To view, create, or modify report-specific runtime options for this report, click **Custom Fields** tab.

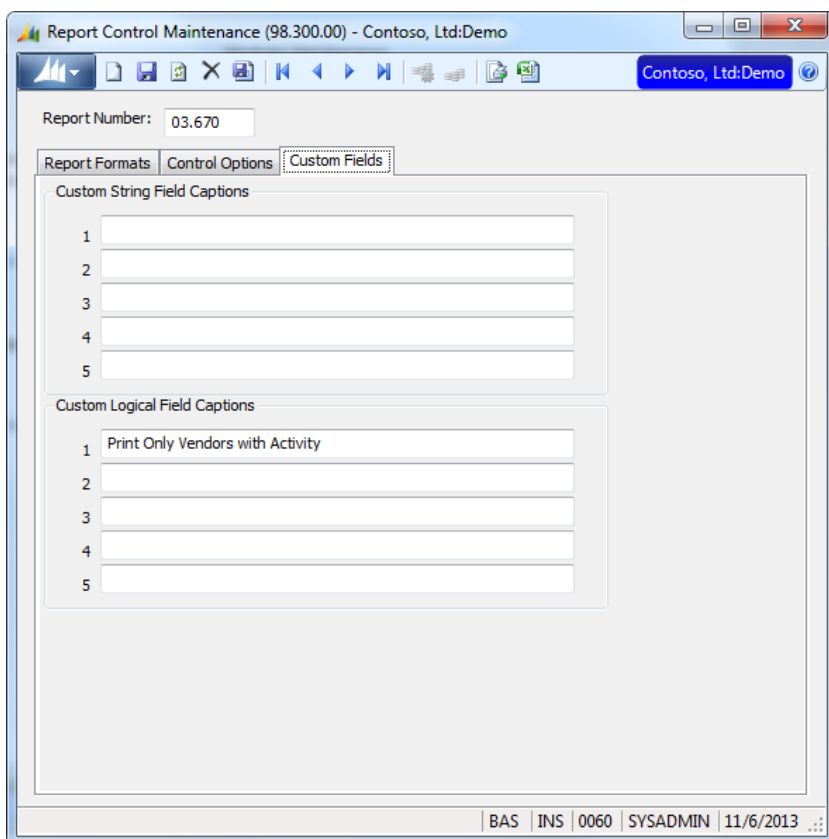


Figure 23: Report Control Maintenance (98.300.00), Custom Fields tab

- To create prompts for text fields that the user can enter values into, type them in **Custom String Field Captions**.
- To create prompts for check box options that the user can choose to turn on or off, type them in **Custom Logical Field Captions**.
- To accept **Custom Fields** changes, click **Save**.

6. To view, create, or modify report processing options, click **Control Options** tab.

The screenshot shows the 'Report Control Maintenance (98.300.00) - Contoso, Ltd:Demo' window. The 'Control Options' tab is active. The 'Report Number' is 03.670. The 'General Reporting Options' section includes: 'Report Date Caption' (Report Date), 'Pre-Process Name' (PB_03670), 'Post-Process Name' (empty), 'Data Source' (Application Database), and checkboxes for 'Disable Banner Prompt' (unchecked), 'Allow Multiple Copies' (checked), and 'Multi-Company Selection Allowed' (checked). The 'Transactional Reporting Options' section includes: 'Reporting Range Prompt' (No Period Number or Date), 'Default Period From' (Accounts Payable Setup), and 'Period Field Name' (empty). The 'Special Form Reporting Options' section includes: 'Print On Special Forms' (unchecked), 'Display Acct/Sub Fields' (unchecked), and 'Document Number Caption' (empty). The status bar at the bottom shows 'BAS | INS | 0060 | SYSADMIN | 11/6/2013'.

Figure 24: Report Control Maintenance (98.300.00), Control Options tab

- In **Report Date Caption**, specify the value that prompts for report date.
- In **Pre-Process Name**, type the name of the pre-process stored procedure or Microsoft SL SDK application, if any.
- In **Post-Process Name**, type the name of the post-process stored procedure or Microsoft SL SDK application, if any.
- In **Data Source**, choose the name of the database type where the data for this report resides.
- If cover pages are not to be allowed for this report, select **Disable Banner Prompt**. This disables the **Cover Page** tab on the ROI window.
- If multiple copies can be created for this report, select **Allow Multiple Copies**. This enables the **Copies** text box on the ROI **Report** tab.
- If data for this report is to be taken from multiple companies, select **Multi-Company Selection Allowed**. This enables the **Company Selection** tab on the ROI window.
- Select an option from the **Reporting Range Prompt**, which determines how reporting dates are handled. The options are as follows:
 - No Period Number – No period or date specified
 - Period to Report – To specify a single period
 - Beg\End Period to Report – To specify a range of periods
 - Calendar Year – To specify a calendar year
 - Validated Period to Post – To specify only a period that is not closed
 - Date to Report – To specify a single date
 - Date Range – To specify a range of dates

- i) If the default period value is to be taken from a setup record, select the one you want in **Default Period From**, or select None.
 - j) In **Period Field Name**, enter the name of the database field the ROI will use to change the RI_WHERE field to produce a period-based report.
If Date to Report or Date Range is selected from the **Reporting Range Prompt** list, the **Period Field Name** field label changes to **Date Field Name**. In **Date Field Name**, enter the name of the database field the ROI will use to change the RI_WHERE field to produce a date-based report.
 - k) If this report is to be printed on special forms instead of plain paper, select **Print on Special Forms**. Selecting this option enables the next two options.
 - l) If the report should have account and subaccount values displayed, select **Display Acct/Sub Fields**.
 - m) In **Document Number Caption**, type text that will identify the type and number of documents being created by this report.
 - n) To accept control option changes), click **Save**.
7. Assign additional report format names and definitions as you want.
 8. Save the record.
 9. Close the screen.
 10. Run the report from the menu to verify the results.

Advanced Reporting Options

Microsoft Dynamics SL Reporting Components

The following components are used by developers to modify the standard Microsoft Dynamics SL reports or to develop custom reports for Microsoft Dynamics SL and add-in applications.

- Application and library software
 - CrystalDecisions.CrystalReport.Engine.dll
 - CrystalDecisions.ReportAppServer.ClientDoc.dll
 - Crystal Reports 2008 runtime files
 - CrpeSol4.dll
 - DynamicsSLFormatting.dll
 - Microsoft.Dynamics.SL.Reporting.dll
 - ROI.exe
 - ROISRS.exe
 - U2LSol4.dll
- Report data files
 - *Filename.rdl* (SSRS)
 - *Filename.rpt* (Crystal)
- Database tables
 - ROIDetail
 - ROIHeader
 - RptCompany
 - RptControl
 - RptRuntime
 - RptExtra

Using advanced ROI options requires an understanding of the core ROI components, their purpose and function, and how they interact. The core group of application and library software, report data files, and database tables and their interaction are explained in the topics that follow.

ROI and ROISRS Software Interaction

The figures and information in this section explain the interaction that occurs between the programs ROI.exe or ROISRS.exe and library software when a report is printed. Each software element described performs a specific set of reporting functions during report generation:

Note: The dynamic link library files (DLLs) described in the topics that follow implement standard Crystal Reports user function libraries. Much of the DLL code is provided by Crystal Reports and is not described in this documentation. To gain an understanding of these DLLs, see information for Crystal Reports developers provided at www.sap.com/solution/sme/software/analytics/crystal-bi/index.html.

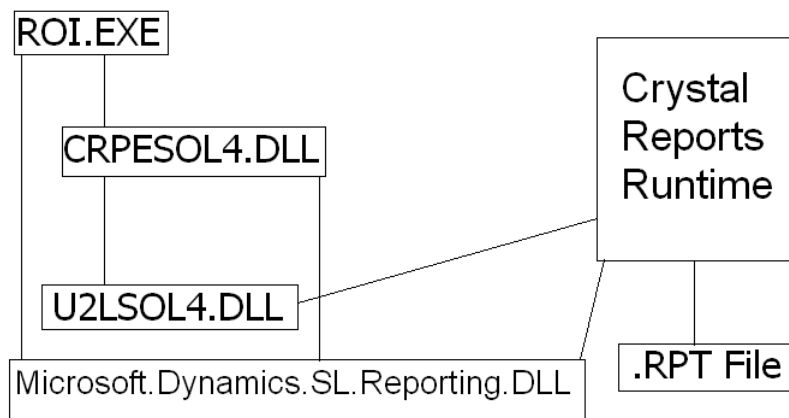


Figure 25: ROI software interaction

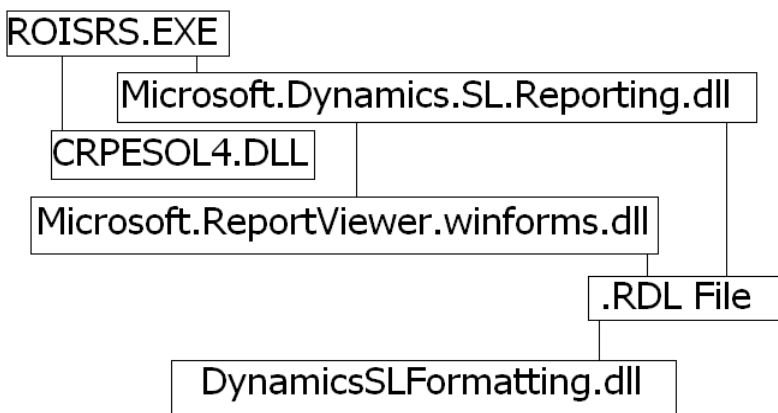


Figure 26: ROISRS software interaction

ROI.exe

The Report Option Interpreter (ROI.exe) is used to create Crystal Reports. It collects the database table information that controls how a report appears when you print it: which records to print on the report, in what order, and where to print them.

ROI.exe calls CrpeSol4.dll to relay its information to Microsoft.Dynamics.SL.Reporting.exe. It also calls Microsoft.Dynamics.SL.Reporting, which calls the Crystal runtime files to obtain report information. This information is used to populate possible values (PV) lists for fields on the ROI screen **Sort** and **Select** tabs.

The ROI lets developers:

- Create a custom ROI window appearance
- Call other processes before or after they print a report

- Develop and maintain different formats of the same report
- Add special data to a report at run time
- Use special functions and formulas in reports
- Print a report in a noninteractive mode
- Make all ROI window information available to other processes
- Add special numbering schemes to reports
- Debug report printing issues
- Modify how reports run

ROISRS.EXE

ROISRS.exe, the Report Option Interpreter for reports generated using SQL Server Reporting Services, calls Microsoft.Dynamics.SL.Reporting.dll to obtain information about a report. This information is used to populate possible values (PV) lists for fields on the ROI screen **Sort** and **Select** tabs.

The ROISRS lets developers:

- Create a custom ROI window appearance
- Call other processes before or after they print a report
- Develop and maintain different formats of the same report
- Add special data to a report at run time
- Use special functions and formulas in reports
- Print a report in a noninteractive mode
- Make all ROI window information available to other processes
- Add special numbering schemes to reports
- Debug report printing issues
- Modify how reports run

For information about how database tables work with ROI and ROISRS to produce Microsoft Dynamics SL reports, see “Database Tables” on page 39.

CrpeSol4.dll

CrpeSol4.dll is a dynamic link library file (DLL) used for communication from ROI.exe to Microsoft.Dynamics.SL.Reporting.dll through U2LSol4.dll to Crystal Reports runtime. The information CrpeSol4.dll communicates is the data found in RptRuntime, RI_PARAM values, and financial setup records. This library is maintained in memory for each report process. You can access it from all client applications. CrpeSol4.dll was created using Microsoft Visual C++.

CrpeSol4.dll stores:

- All RIParams
- Many setup records (stored as RIParams)
- A copy of each **Sort** and **Select** tab grid line from ROI.exe

U2LSol4.dll

U2LSol4.dll, a user function library file, is specific to Crystal Reports. It contains functions needed to complete the processing of and add functionality to Microsoft Dynamics SL reports. It can perform many tasks, such as manipulate data or obtain information from memory space in CrpeSol4.dll and return it to a report. U2LSol4.dll is called by Microsoft.Dynamics.SL.Reporting.dll if the formula is included in the report at design time.

U2LSol4.dll was created using Microsoft Visual C++. For more information about Visual C++, see msdn.microsoft.com/en-us/vstudio/hh386302.

For more information, see “U2LSol4.dll Functions” on page 74.

Microsoft.Dynamics.SL.Reporting.dll

The main purpose of Microsoft.Dynamics.SL.Reporting.dll is to process runtime changes to a report file. During ROI or ROISRS operations, Microsoft.Dynamics.SL.Reporting.dll

- Modifies the WHERE clause in the report’s SQL statement to match values entered on the ROI screen.
- Produces the report and routes it in local mode to the desired destination: display, file, or printed.
- Creates the database connections for the report.
- Changes the report’s groupings based on input from the ROI screen.
- Modifies the font of the report.

DynamicsSLFormatting.dll

DynamicsSLFormatting.dll is used during SSRS report processing. It contains user functions for SSRS reports. This resembles the U2LSol4.dll file used by Crystal Reports. For information, see “User-Defined Functions Specific to SSRS Reports” on page 105.

ROI Command Line Parameters

Note: The following information applies to both the ROI.exe and ROISRS.exe.

To run reports from the command line you need to use specific parameters. The following is an example of a full set of parameters:

- ROI [{ReportNbr}/RUN]
- [{ReportFormat}/FORMAT]
- [{Condition}/WHERE]
- [/PSCRN]
- [/DEBUG]
- [{Template ID}/TEMPLATE]
- [{Report Date}/RPTDATE]
- [{Filename}/EXPFILENAME]
- [{Format ID}/EXPFORMAT]
- [{BatchNumber}/BATCH]
- [{Doc Number}/DOCNBR]

Example:

```
ROI.EXE 0 01660/RUN 01660/FORMAT 0/EXPFORMAT c:\01660.pdf/EXPFILENAME
```

When you run the ROI program from the command prompt, be aware of the following:

- The first parameter passed to the ROI **must** be a zero. When the ROI is called using Callapplic(), Launch(), or Callapplicwait(), the parameter is added for you.
- Parameters should be separated by the TAB character. However, when you call the ROI from the command prompt, this is not possible. In this case, spaces are used to separate parameters.

Specific Parameters

BATCH

Specifies the batch number that is the basis for the report. **BATCH** is available only for selected reports that show the batch number field on the ROI window.

Example:

```
882376/BATCH
```

DEBUG

Instructs the program to run in debug mode. The ROI will display messages about what is occurring. Although **DEBUG** is unrefined, it can be helpful at times. It does not have a value associated with it.

DOCNBR

Specifies the document or reference number that is the basis for the report. This option is available only on selected reports that show the field on the ROI window.

EXPFILENAME

Specifies the name of a file to which the report will be exported. **EXPFILENAME** does not support file names that contain spaces. The ROI considers spaces as separators

Example:

```
C:\DATA\01650.pdf/EXPFILENAME
```

EXPFORMAT

Specifies the code for the file format in which the report will be exported. For possible values, see “RI_FILETYPE.”

Example:

```
0/EXPFORMAT
```

In this example, the report will be written to a PDF file.

FORMAT

A report format name for the report ID must be specified with **FORMAT**. This is the report file name (without the extension), **not** the name you see in the ROI window.

Example:

```
01650MC/FORMAT  
01610A/FORMAT
```

PSCRN

Instructs the program to open a print preview window instead of printing the report. This option does not have a value associated with it.

RPTDATE

Specifies the value in the ROI report date field when you run the report.

Example:

```
01/23/2005/RPTDATE
```

RUN

A valid Microsoft Dynamics SL report ID must be specified with **RUN**.

Example:

```
01650/RUN  
01660/RUN  
03670/RUN
```

TEMPLATE

Instructs the ROI program to apply the specified template to the screen before processing the **RUN** command.

Example:

```
SmallChart/TEMPLATE
```


WHERE

Specifies special selection criteria to apply to the report. For SSRS reports, use Transact-SQL syntax rules when you construct **Where** clauses. For Crystal Reports, the clauses need to follow Crystal syntax rules, including delimiting field names with braces ({}) and using qualified field names (<Table name>.<Field name>).

SSRS Example:

```
Account.Acct<'9000'/WHERE  
Account.Acct between '1000' and '9000'/WHERE
```

Crystal Examples:

```
{Account.Acct}<'9000'/WHERE  
{Account.Acct} in '1000' to '9000'/WHERE
```

Report File Search Method

The ROI must search for the correct report file. ROI.exe searches for files that have the extension of .rpt, while ROISRS.exe searches for files that have the extension of .rdl. To find the correct file, the ROI looks for the Microsoft Dynamics SL report file in:

1. Microsoft Dynamics\SL\Applications\Usr_Rpts
2. Microsoft Dynamics\SL\Applications\?? (where ?? is the module)

Note: This method takes into account cases in which users customize reports and put them in the Usr_Rpts subdirectory.

Date Usage

Any date fields included on a report should have their formatting set to Use Windows Default. Several date fields in the RIParam fields are already formatted correctly by the ROI.

A user-defined function named DateFmt() reformats date fields to match the current Windows setting.

Understanding Pre-Processes

What Are Pre-Processes?

Microsoft Dynamics SL pre-processes are applications that run before a report is printed. Pre-processes can increase the performance of reports by pulling data from database tables, converting the data, and storing it in temporary work tables.

A pre-process can also be a Microsoft SL SDK application or a stored procedure. The ROI first looks for a stored procedure with the same name as the pre-process. If it does not find a stored procedure, it tries to run a program (an .exe file) with the same name.

In addition, you can use a pre-process to change report selection criteria at run time by modifying the RI_WHERE field in the RptRuntime record.

When Are Pre-Processes Called?

Pre-processes are called as the last item processed before the report is displayed, exported, or printed.

Debug Mode

ROI.exe, ROISRS.exe, and Microsoft Dynamics.SL.Reporting.dll are provided with a Debug mode that can help developers debug problems in reports and the reporting process. Debugging provides the following information:

- The RI_ID
- Indications that pre-processes are running
- The option to print or not to print the report (after all ROI processing is complete)
- The report query just after opening the report
- The report table names, before and after being changed (based on the database name)
- Printer information

Starting Debug Mode

Create a file named Debug.tlb in the Microsoft Dynamics\SL\Applications folder. Debug mode starts at the same time as ROI.

Pre-process Cancellation of Report Generation

While performing its tasks, a pre-process may encounter conditions that will prevent proper report generation. The pre-process can cancel report generation by passing information back to the ROI so that it can abort the process. How the pre-process passes information back to the ROI depends on the pre-process type:

- SQL Server stored procedure
- Microsoft SL SDK application

SQL Server Stored Procedure Pre-process

In order for a SQL Server stored procedure to cancel report generation, the stored procedure must raise an error using the RAISERROR T-SQL statement. The statement must use an ad-hoc message format. This ensures a standard error code for the ROI to read and interpret. The message value should be a formatted string containing:

- The number of an informational message identifying the reason for canceling report generation.
- Substitution variables for use in the message.

If canceling report generation is an expected occurrence, rather than an exception to the process, the message value should be None. The severity and state values of the error can be any valid values.

The stored procedure RAISERROR T-SQL statement uses this format:

```
RAISERROR('message', severity, state)
```

Examples:

```
RAISERROR('6999, Account.acct, SYSADMIN', 3, 1)
```

This statement cancels report generation and issues message 6999, as found in the Messages.csv file. This statement includes the substitution variables Account.acct and SYSADMIN.

```
RAISERROR('None', 3, 1)
```

This statement cancels report generation but does not issue a message.

Microsoft SL SDK Application Pre-processes

Microsoft SL SDK applications can return parameters to calling programs using APPLICRETURNPARMS as the first parameter in the screenexit() call. The first return string parameter must be one of two values:

- ROI_ABORT_REPORT
- ROI_EXECUTE_REPORT

The second parameter should be a formatted string containing:

- The number of the informational message identifying the reason for canceling report generation.
- Substitution variables for use in the message.

Database Tables

ROI.exe and ROISRS.exe obtain information that controls how reports appear when printed: which records to print on a report, in what order, and where to print them. For every different report format, the database tables of each are controlled by a single RptControl record in the system database.

- RptRuntime database tables control what ROI.exe does during the report printing process. For information on the RptRuntime records, see “RptRuntime”.
- ROIHeader and RoiDetail database tables control what is printed on cover page reports. For information on the ROIHeader and RoiDetail records, see “ROIHeader” and “ROIDetail.”
- RptCompany database tables control the company information printed in reports.
- RptExtra database records can pass information to the ROI from a calling application or a pre-process.

RptControl

The RptControl record controls how each ROI window looks and acts. The contents of RptControl determine which tabs are active in the ROI window, as well as what happens when you click the **Print** button. An RptControl record must exist for each report.

Following is a list of RptControl record fields. Included with each is a description of what the field contains, a possible values (PV) list, and what it causes the ROI to do.

AcctSub As integer

Tells the ROI whether or not to display the account and subaccount fields. This setting is evaluated only when the MODULE field of this record is also set.

Value	Description
>0	Display the fields on the ROI window
0	Do not display the fields

Default Information

When account and subaccount are displayed, their default is the CHKACCT field from the setup record of the module specified by MODULE.

CopiesAllowed As integer

Tells ROI whether or not to display **Copies**, **Beginning Page**, and **Ending Page** in the ROI window.

Value	Description
>0	Display the fields on the ROI window
0	Do not display the fields

Default Information

When displayed, defaults to the value stored in this field.

CpnyFlag As integer

Currently not used.

DBType As String * 1

Tells the ROI and Microsoft Dynamics.SL.Reporting.exe the database type associated with the report, either the current application database or the current system database.

Value	Description
S	System database
A	Application database

DocPrompt As String * 20

Value	Description
MASK	Tell ROI the user has the option to print report masks <anything>

When this field is set, the **Print Mask** and **Print Preview** buttons become visible. If the field value here is not MASK, **Batch Number** and **Document Number** become visible.

LongAnsCaption(0 To 4) As String * 41

Poses up to five additional questions (report options) to the user running the current report. User responses are written into the RptRuntime record. If any of these values are set, the following occurs:

- The **Option** tab is enabled.
- A control for input of a long answer to the question (option) is activated.

Module As String * 2

Specifies the Microsoft Dynamics SL module associated with the report. This setting helps determine which setup record to use when defaulting values on the ROI window.

MCSlection As Integer

Tells the ROI whether or not the current report is a multi-company report. If the report is multi-company, the **Company Selection** tab appears in the ROI window. See “Company Selection Tab Settings.”

Value	Description
0	Do not display the tab
1	Activate the Company Selection tab in the ROI window

NoBanner As Integer

Tells the ROI whether or not to enable the **Cover Page** tab, so users can print the cover page report before printing the current report.

Value	Description
1	Activate the Cover Page tab in the ROI window
2	Do not display the tab

PerFieldName As String * 41

The only time this field is used is when **PerPrompt** is 1 or 2. When used, this field must contain a database field name that contains period information (for example, GLTran.Pernbr). See “PerPrompt As Integer.”

PerPrompt As Integer

Value	Description
6	Refers to date range reporting: <ul style="list-style-type: none"> • Make the Beginning Date and Ending Date fields visible • Set the RptRuntime.RI_WHERE = PerFieldName>= "Beginning Date and PerFieldname <= Ending Date"
5	Refers to date reporting: <ul style="list-style-type: none"> • Make the Beginning Date field visible • Set the RptRuntime.RI_WHERE = PerFieldName = "Beginning Date"
4	Refers to a period that has been closed: <ul style="list-style-type: none"> • Make Beginning Period visible • Set the RptRuntime.RI_WHERE = "PerFieldName like Beginning Period" • The value you enter here must be a closed period
3	Refers to calendar year reporting: <ul style="list-style-type: none"> • Make the Calendar Year input field visible • Put the current system date into the RptRuntime.batnbr field • Set the RptRuntime.RI_WHERE = ""
2	Refers to period range reporting: <ul style="list-style-type: none"> • Make the Beginning and Ending Period input fields visible • Set the RptRuntime.RI_WHERE = "PerFieldname is between the beginning and Ending period"
1	Refers to period reporting: <ul style="list-style-type: none"> • Make Beginning Period visible • Set the RptRuntime.RI_WHERE = "PerFieldName like Beginning Period"
0	Removes calendar and period reporting: <ul style="list-style-type: none"> • Do not display any period or calendar year input fields • Set the RptRuntime.RI_WHERE = ""

PrintOnForms As String * 1

A shortcut for making the **Print Mask** control visible in the ROI window.

Value	Description
Y	Make the Print Mask control visible; make the Print Preview button invisible

RecType As String * 1

Always set to S for Standard Reports. *Do not change this setting unless you are creating a custom report!* The record type value to use for custom reports is C.

ReportDateCaption As String * 20

Alternate text for the default **Report Date** label. Common alternatives include **Check Date** and **Statement Date**. In the ROI window, there is a field labeled **Report Date** that displays the date when the report is printed. You can change this label by modifying this field's contents.

ReportNbr As String * 5

The Microsoft Dynamics SL window number for the current record (for example, 01650, 04600, 03670). When users select a report from the menu, this window number is passed to ROI. ROI uses it to retrieve the correct RptControl record.

RunAfterApplic As String * 12

Informs the ROI that a post-process program or SQL statement should be run after a report finishes printing (for example, calls other processes after printing the report). The contents of this field and the database type determine whether or not the post-process is an SQL statement. The ROI observes the following rules:

- If the database type is Microsoft SQL Server, the value can either be an SQL statement or a program.
- If a stored procedure with the same name as the value exists, run it. Otherwise, run the program with the same name as the value.
- No .exe is required for this field.

RunBeforeApplic As String * 12

Informs the ROI that a pre-process program or SQL statement should be run before the report is printed (for example, calls other processes before printing the report). The contents of this field and the database type determine whether or not the pre-process is an SQL statement. The ROI observes the following rules:

- If the database type is Microsoft SQL Server, the value can either be an SQL statement or a program.
- If a stored procedure with the same name as the value exists, run it. Otherwise, run the program with the same name as the value.
- No .exe is required for this field.

ShortAnsCaption(0 To 4) As String * 41

Poses up to five yes or no type questions to users running the current report. User answers to these questions are written into the RptRuntime record. If any of these values are set, the following occurs:

- The **Option** tab becomes enabled
- A control for input of a short answer to the question (option) is activated

This control allows you to create a custom ROI window appearance, call other processes before and after printing a report, develop different formats for the same report, and ask users additional questions (report options).

RptFormat

This table contains the format information that is used by the ROI and ROISRS screens.

DisplyIndex As Integer

Determines the order of the formats as shown in the drop down box on the ROI or ROISSRS screen.

FileName As String * 30

The name of the report file.

FormatName As String * 30

Format names for unlimited number of report files. The report to be printed resides in the file name maintained here.

ReportNbr As String * 5

The Microsoft Dynamics SL window number for the current record (for example, 01650, 04600, 03670). When users select a report from the menu, this window number is passed to ROI or ROISSRS. ROI/ROISSRS uses it to retrieve the correct RptControl record.

RptRuntime

The RptRuntime record controls what ROI.exe does during report printing (for example, after you click the **Print** button). Each report has an RptRuntime record, and the information in each RptRuntime record is available to other processes.

AccessNbr As String * 5

Access number of the current session set by ROI.

Acct As String * 10

An account number obtained from the ROI window.

Banner As String * 1

A value set to T if a cover page report is to be printed before printing the report.

BatNbr As String * 10

Batch number value obtained from the ROI window.

BegPerNbr As String * 6

Beginning period number value obtained from the ROI window. If one or two period fields appear on the ROI window, this field is populated from a setup record.

BusinessDate As String * 10

Business date value obtained from the kernel (set when you choose **Business Date** on the application toolbar).

CpnyID As String * 10

Company name value obtained from the GLSetup record.

CmpnyName As String * 30

Company name value obtained from the GLSetup record.

ComputerName As String * 21

Computer name obtained from the Kernel.

DatabaseName As String * 20

The name of the database used to run the report. This value is used to modify the table names in the .rpt file at runtime. It determines which ODBC database to use for printing the report.

DocNbr As String * 10

Document number value obtained from the ROI window.

EndPerNbr As String * 6

Ending period number value obtained from the ROI window. If one or two period fields appear on the ROI window, this field is populated only when there are two period fields.

LongAnswer(0 To 4) As String * 80

Up to five values obtained from the **Option** tab in the ROI window. These values are the answers to the long questions (report options) on the **Option** tab. The questions are controlled by the developer-defined RptControl record.

NotesOn As Integer

A value obtained from the ROI window that indicates whether or not to print notes (from the SNote table) on the report.

Value	Description
0	Do not print notes
1	Print notes

Note: The report must be customized (for example, the SNote fields must be added to the report) in order for this setting to have any effect. See “Adding SNote Fields to Crystal Reports.”

PerNbr As String * 6

The current period number associated with the report. The ROI populates this field based on the RptControl Module field. The ROI accesses a setup record at the specified module and copies the period number to this field.

RepBegDate As String * 10

The value of this SmallDateTime field is obtained from **Begin Date** on the ROI window. It is used to update the **RI_WHERE** field to generate a report based on a date. Store the date value in a MM\DD\YYYY format.

RepEndDate As String * 10

The value of this SmallDateTime field is obtained from **End Date** on the ROI window. The values in this field and the **RepBegDate** field are used to update the **RI_WHERE** field to generate a report based on a range of dates. Store the date value in a MM\DD\YYYY format.

ReportDate As String * 10

Report date obtained from the ROI window. This value/field can have different captions, based on the RptControl ReportDateCaption field.

ReportFormat As String * 30

Report name obtained from the FormatName field in the RptFormat record. This is the descriptive name for the format, as the user sees it in the ROI window list.

ReportName As String * 30

Report file name obtain from the FileName field in the RptFormat record. This is actually the file name in which the report is stored. The file name represented here does not include the extension of the file nor its location. The ROI will determine the file name and extension.

ReportNbr As String * 5

Report number obtained from the RptControl record.

ReportTitle As String * 30

The same value as RI_REPORT.

RI_BEGPAGE As Integer

Beginning page number of the first report page to be printed. This value is obtained from the ROI window.

RI_CHKTIME As String * 1

Currently not used.

RI_COPIES As Integer

Number of report copies to be printed. This value is obtained from the ROI source.

RI_DATADIR As String * 21

Currently not used.

RI_DICTDIR As String * 21

Currently not used.

RI_DISABLEDSD As SmallInt

A value obtained from the ROI window that indicates whether or not to publish data items on the report to Microsoft® SharePoint® sites based on Doc Share settings.

Value	Description
0	Publish data items to SharePoint sites via Doc Share
1	Do not publish data items to SharePoint sites via Doc Share

RI_DISABLEQS As SmallInt

A value obtained from the ROI window that indicates whether or not to send data items on the report to customers, employees, projects, and vendors based on their Quick Send preferences.

Value	Description
0	Send data items via Quick Send
1	Do not send data items via Quick Send

RI_DISPERR As String * 1

Currently not used.

RI_ENDPAGE As Integer

Page number of the last report page to be printed. This value is obtained from the ROI window.

RI_ID As Integer

Unique ID for the current RptRuntime record. This number identifies all ROIHeader and ROIDetail records that go with the current report run, as well as all RIParams and work tables that are created by pre-processes.

RI_INCLUDE As String * 1

Currently not used.

RI_LIBRARY As String * 80

Currently not used.

RI_NOESC As String * 1

Currently not used.

RI_OUTAPPN As String * 1

Currently not used.

RI_OUTFILE As String * 80

Name of the file where report output is sent if **Print To File** is selected in *Printer Options* (98.220.00).

RI_PRINTER As String * 1

Name of the printer to receive the report output.

RI_REPLACE As String * 255

Text needs to be supplied here.

RI_REPORT As String * 30

Caption obtained from the ROI form.

RI_STATUS As String * 1

Currently not used.

RI_TEST As String * 1

If set to Yes, specifies that a mask should be printed when the user clicks **Print** in the ROI window. Clicking **Print** calls Microsoft.Dynamics.SL.Reporting.exe, and one mask is printed. After this, Microsoft.Dynamics.SL.Reporting.exe asks if another mask should be printed. Each time the user answers Yes, Microsoft.Dynamics.SL.Reporting.exe prints the mask.

RI_WHERE As String * 255

Part of an SQL statement created by the ROI. The ROI takes information from the ROI window and compiles it into an SQL WHERE clause. RI_WHERE information includes:

- Select criteria from the **Select** tab.
- Period information. Run the report on the selected period or period range.

RI_WPORT As String * 40

Printer port information: lpt1 or a network path.

RI_WPTR As String * 40

The name of the printer used by Windows.

RI_WTITLE As String * 30

A value set to RI_REPORT.

SegCustMask As String * 16

Flex definition mask for the customer ID. The ROI sets this value based on the Flex definitions stored in the database.

SegCustTitle As String * 16

Flex definition title for the customer ID. The ROI sets this value based on the Flex definitions stored in the database.

SegInvenMask As String * 26

Flex definition mask for the inventory ID. The ROI sets this value based on the Flex definitions stored in the database.

SegInvenTitle As String * 26

Flex definition title for the inventory ID. The ROI sets this value based on the Flex definitions stored in the database.

SegSubMask As String * 38

Flex definition mask for the subaccount. The ROI sets this value based on the Flex definitions stored in the database.

SegSubTitle As String * 38

Flex definition title for the subaccount. The ROI sets this value based on the Flex definitions stored in the database.

SegVendMask As String * 16

Flex definition mask for the vendor ID. The ROI sets this value based on the Flex definitions stored in the database.

SegVendTitle As String * 16

Flex definition title for the vendor ID. The ROI sets this value based on the Flex definitions stored in the database.

ShortAnswer(0 To 4) As String * 10

Up to five values obtained from the **Option** tab in the ROI window. These values are the answers to the short questions (report options) on the **Option** tab. The questions are controlled by the developer-defined RptControl record.

Sub As String * 24

Subaccount value obtained from the ROI window.

SystemDate As String * 10

String representation of the computer's current system date.

SystemTime As String * 7

String representation of the computer's current system time.

UserId As String * 47

ID of the current Microsoft Dynamics SL user.

ROIHeader

The ROIHeader record contains cover page report information. If the user chooses to print the cover page report at runtime, the ROI uses this information, as well as the information in ROIDetail record, to generate the report.

Format As String * 30

Format (file name) of the report currently printing.

NamedDefault As String * 8

Currently not used.

RI_ID As Integer

Unique number assigned to the user running a report in a system database. This ID is used to collect all information needed for the report. RptRuntime, ROIHeader, and ROIDetail share this ID.

Comments As String * 1900

Storage area for text entered on the report as the cover page banner. This value resides in an RI_PARAM named RI_COMMENT at runtime so the report can use it.

ROIDetail

The ROIDetail record, along with the ROIHeader record, controls what is printed on cover page reports. ROIDetail also causes Microsoft.Dynamics.SL.Reporting.exe to modify the report's record sorting and record selection specifications.

Users create ROIDetail records using the **Select** and **Sort** tabs in the ROI window. Values selected from the widow are stored in the database for use by the cover page report. The records are also copied into CrpeSol4.dll memory so Microsoft.Dynamics.SL.Reporting.dll can reference and use the information to modify the report at runtime.

CriteriaLVal As String * 100

Literal value used to compare.

CriteriaOp As String * 11

Operators used in compare operations.

- Begins with
- Between
- Contains
- Equal
- Greater than
- Greater than or equal to
- In
- Is null
- Less than
- Less than or equal to
- Not between
- Not contains
- Not equal
- Not in
- Is not null

Field As String * 41

Name of a report field against which the CriteriaLVal is compared, using the CriteriaOp.

LineID As Long

Unique field set by the kernel.

LineNbr As Integer

Unique field set by the kernel.

Operator As String * 3

Determines if two WHERE conditions are joined using the Boolean AND or OR operators.

Value	Description
AND	Join the conditions using the AND operator
OR	Join the conditions using the OR operator

PageBreak As Integer

If the condition is true, place a page break in the report whenever the value of the field denoted by ROIDetail.Field changes. Page break information is included on cover page reports.

RI_ID As Integer

Unique number assigned to the user running a report in a system database. This ID is used to collect all information needed for the report. RptRuntime, ROIHeader, and ROIDetail—as well as RIParam information—share this ID.

SortAscend As Integer

Determines if the report is to be sorted in ascending or descending order.

Value	Description
0	Sort descending
>0	Sort ascending

SortNbr As Integer

Determines sort order: which is most important, second most important, etc.

TotalBreak As Integer

If the condition is true, place a total break in the report whenever the value of the field denoted by ROIDetail.Field changes. Total break information is included on cover page reports.

Utilizing ROISelect

This topic describes the ROISelect feature and explains how to use it. It assumes you are familiar with the basics of how to modify reports, add formulas to a report, and use the ROI.

ROISelect is a powerful tool for report creators. It allows you to specify which fields appear in possible values (PV) lists on the ROI and ROISRS **Sort** and **Select** tabs when you are printing reports. This can prevent you from sorting or selecting fields that are not intended for this purpose.

How to Use ROISelect

To use ROISelect, specify list fields by creating a formula called ROISelect in the report. In the formula, list all fields to be included in the sort and select PV options. Here are details to assist you:

- The formula entered for ROISelect must follow this format:

```
Include,ReportRecord.Field,{DBRecord},{LogicalRecord} or
Exclude,ReportRecord.Field,{DBRecord},{LogicalRecord}
```

- Fields in {} are optional. If they are not used, the commas separating **ReportRecord.Field**, **DBRecord** and **LogicalRecord** are also optional
- A space can be entered after any comma for easier readability.
- A single ROISelect formula can have several include or exclude statements (each section separated by a semicolon). For example:

```
Include,APTran;Include,APDoc.batnbr;Exclude,APTran.acct;
```

ROISelect Formula Elements

- Include** or **Exclude** – Describes whether to add or remove PV fields from the sort or select list. Excludes are typically the last sections of an ROISelect formula. When the ROI encounters an ROISelect formula, the current PV list is cleared and a new PV list is defined by the fields listed in the formula.

For example, a PV list might **Include** all APTran fields, but **Exclude** the APTran.acct field. The **Include** needs to occur first to add all APTran fields to the PV list so that **Exclude** can remove the APTran.acct field from that list (if **Exclude** is done first, it tries to remove APTran.acct from an empty PV list). The formula for this example is:

```
Include,APTran;Exclude,APTran.acct;
```

- ReportRecord.Field** – Identifies the fields to include or exclude in the PV list. Typically, these are items such as APTran.acct or APDoc.batnbr. It is possible to include or exclude all fields within a record by using a format such as APTran.*, or simply APTran. In the previous example, the formula could also be entered as

```
Include,APTran.*;Exclude,APTran.acct;
```

- DBRecord** – An optional field that is used when the ReportRecord name is an alias for the database record name. For example, some reports might have the alias APDoc2 as the ReportRecord name when the record name in the database is APDoc. For the formula `Include,APDoc2,APDoc`; all fields in the APDoc record would appear in the PV window as APDoc2 fields.
- LogicalRecord** – An optional field that provides a more user-friendly description of the record or field names. The description appears in the Field Description column of the PV window. For example, the formula `Include,APTran.TranDate,,Transaction Date`; appears in the PV list as

```
Record.FieldName:    aptran.trandate
Field Description:   Transaction Date
```

ROISelect Formula Examples

The *AP Transactions* (03640b) report, by default, shows all fields from APDoc, APTran, and RptCompany in the sort and select PV lists. The following are examples of adding ROISelect to the report to limit the tables and fields that appear in the PV lists:

- These expressions include all fields from only the APTran table:


```
Include,APTran;
Include,APTran.*;
Include,APTran,,;
```
- These expressions display **Batch Number, Vendor ID, and Transaction Type** from the APTran table:


```
Include,APTran.BatNbr;APTran.VendID;APTran.TranType;
Include,APTran.BatNbr,,Batch Number;APTran.VendID,,Vendor
ID;APTran.TranType,,Transaction Type;
```
- These expressions display **Batch Number, Vendor ID, Reference Number, and Terms ID** from the APTran and APDoc tables:


```
Include,APTran.BatNbr;APTran.VendID;APDoc.RefNbr;APDoc.Terms;
Include,APTran.BatNbr,,Batch Number;APTran.VendID,,Vendor
ID;APDoc.RefNbr,,Reference Number;APDoc.Terms,,Terms ID;
```

Note: The semicolon after APTran is not required since the formula contains double quotation marks and only one Include statement.

ROISelect Notes

- This feature does not allow modifications to the current possible values (PV) list. If a report contains a ROISelect formula, the current PV list is cleared and only those fields specified in the ROISelect formula appear in the PV list.
- While ReportRecord.* can be used, the logical record, if used, shows for all fields. This means you cannot have descriptions for each field if you use this approach.

Implementing ROISelect for Crystal Reports

- The formula must be called ROISelect.
- The **Include** rule for ReportRecord.* obtains all fields for that database record, not all fields for that record in the report.
- Crystal Reports has an expression limit of 255 bytes. This limit can be reached with a large list.
- When you save the ROISelect formula, you may receive the message, "There is an error in this formula. Do you want to save it anyway?" Click **Yes** to ignore the message. To eliminate the error, add double quotation marks around the text of the formula (for example, "Include,APTran").

Implementing ROISelect for SQL Server Reporting Services

- The code function must be called ROISelect and should look similar to:

```
Function ROISelect() as String
ROISelect = "Include,Customer.Custid,,Customer ID;Customer.Zip,,Zip Code"
End Function
```

The syntax is very specific.

- The uppercase and lowercase letters in *ROISelect* must be exactly as shown.
- Do not insert a space after *t* in ROISelect and before (.
- In the second line of code, be sure to insert a space between *t* in ROISelect and =.
- The function must return a string.
- The **Include** rule for ReportRecord.* obtains all fields for that database record, not all fields for that record in the report.

Using Crystal Reports

With Crystal Reports, you can modify the standard reports or add custom reports that you have developed. You can also provide access to new reports from within the menu system. That way, users are not required to print these reports from within the Report Designer.

You can add a new report to the available formats of another report, or you can insert it as a new item on the Microsoft Dynamics SL menu. If you add the report to formats for another report, users can then select it from the **Report Format** list in the report's ROI window. If you add the report to a Microsoft Dynamics SL menu, users can access it from the menu's **Reports** list.

It is assumed that you have purchased, installed, and registered Crystal Reports, and that you have the requisite knowledge of Crystal Reports and database tables. Knowledge of a database modification utility such as SQL Query Analyzer or ISQL/W for Microsoft SQL Server is also required.

Note: Some of the procedures that follow contain SQL statements you will run against your system database. In all instances, uppercase (capital) letters are very important. Save your SQL statements in a script file so that they are readily available in the future.

Creating a New Report from an Existing One with Crystal Reports

The example below explains how to create a new report by modifying an existing one using Crystal Reports. This example uses the combined totals format of the General Ledger *Trial Balance* (01.610.00) report.

To create a new report from an existing one using Crystal Reports:

1. Open Crystal Reports.
2. On the File menu, click **Open**.
3. Select the 01610a.rpt file located in \Microsoft Dynamics\SL\Applications\GL.
4. The message "This report could not be opened for writing. Any changes must be saved to a new file" appears. Click **OK**, and the report opens in the **Design** tab.
5. Before making changes, select **Save As** on the File menu. Save the file as 01newv1.rpt and store it in \Microsoft Dynamics\SL\Applications\Usr_Rpts.

Note:

- Keep all customized Microsoft Dynamics SL reports in the Usr_Rpts folder.
 - Including the *01* prefix in the report name is suggested for this General Ledger report. Use the appropriate prefix for the module that is associated with the report you are customizing (for example, *08* for an Accounts Receivable report).
6. Modify the report as desired. In this example, two changes are involved.
 - a) The company ID is added to the report header next to the company name using the format name "With Company ID."
 1. Right-click on the **@Prt_CmpnyName** field, and then select Edit Formula. *Formula Editor* opens.
 2. Change the formula to include the company ID.


```
TRIM ({vr_01610A.Rpt_Company_CpnyName} + "(" +
              {vr_01610A.AcctHist_CpnyId} + ")")
```
 3. Click **Save and Close**.
 - b) Ending balances with negative values are surrounded by parenthesis using the format name "Negative in Parens."

1. Right-click on **Ending_Balance**, and then select Format Field. The *Format Editor*, **Number** tab appears.
 2. Click **Customize**. The *Custom Style*, **Number** tab appears.
 3. Select (123) from the **Negatives** list.
 4. Click **OK** twice to return to the **Design** tab.
7. Choose **File | Save As** again, and save this report in Microsoft Dynamics\SL\Applications\Usr_Rpts as 01newv2.rpt. There are now the two formats of the new report, 01newv1 and 01newv2.

To insert a report as an additional format of another report, see “Adding a New Report Format” on page 61.

Adding a Report to a Menu

Be sure that you have the appropriate access rights to use the screens in the following steps. See the System Management Help or user's guide for additional information about the screens in these procedures.

Note: We recommend that you create a backup copy of your database before you proceed.

Create a New Report Control Record

To add reports to the Report Control table (RptControl), you can use *Report Control Maintenance* (98.300.00) or SQL statements. A record must exist in RptControl for each Microsoft Dynamics SL report. See "Maintaining Report Control Records" on page 25 or the System Manager Help or user's guide for more information about *Report Control Maintenance* (98.300.00).

To add a report control record using *Report Control Maintenance* (98.300.00):

1. In the Microsoft Dynamics SL window, click **Administration**.
2. Click **Report Control Maintenance** in the System Manager window. *Report Control Maintenance* (98.300.00) appears.

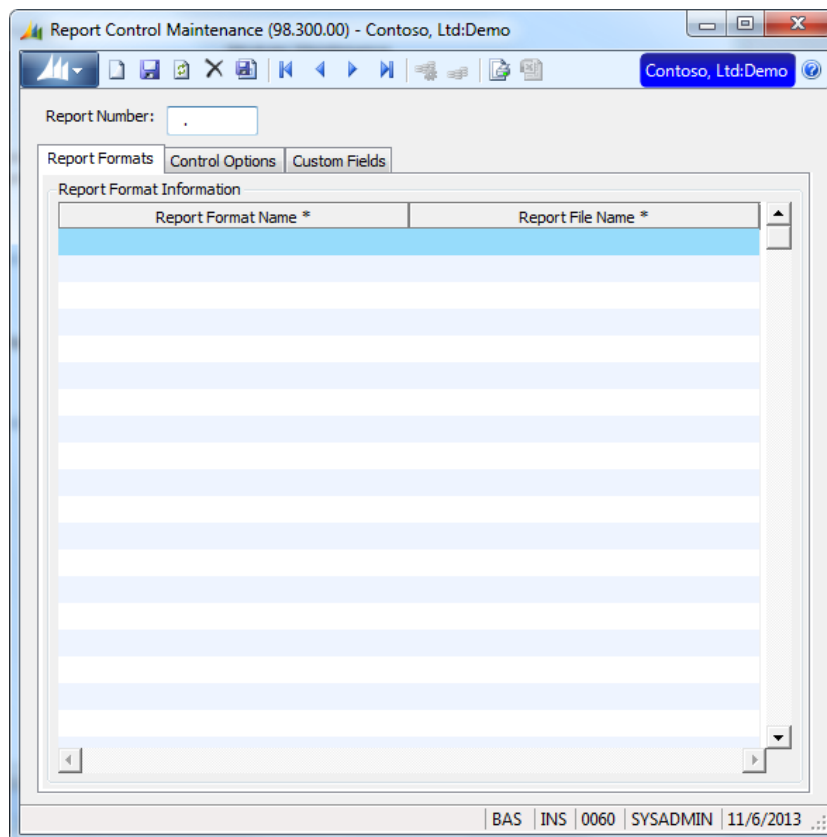


Figure 27: Report Control Maintenance (98.300.00)

3. In **Report Number**, type the number of your custom report without the .rpt or .rdl file extension. For example, the report number is entered as 01-999 if the report file name is 01999.rpt or 01999.rdl.
4. In **Report Format Name**, type a name that describes the custom report format.
5. In **Report Format**, type the report format identifier. For report number is 01.999, for example, the report format identifier is entered as 01999.
6. Save and close the screen.

In *Report Control Maintenance* (98.300.00), the **Report Number**, information on the **Control Options and Custom Fields** tabs are all stored in the RptControl table. The formats are all stored in the RptFormat table.

To add reports to the Report Control table using SQL statements:

1. Open a database modification utility, log on to the appropriate system database, and run the following statement to review the RptControl record:

```
SELECT * FROM RptControl WHERE ReportNbr = '01NEW'
```

2. Add the new report with its two formats to the RptControl and RptFormat tables by typing and running the following SQL statement:

```
INSERT INTO RptControl
(AcctSub, CopiesAllowed, CpnyFlag, DBType, DocPrompt,
LongAnsCaption00, LongAnsCaption01, LongAnsCaption02, LongAnsCaption03,
LongAnsCaption04, MCSelection, Module, NoBanner, PerFieldName, PerPrompt,
PrintOnForms, RecType, ReportDateCaption, ReportNbr, RunAfterApplic,
RunBeforeApplic, ShortAnsCaption00, ShortAnsCaption01, ShortAnsCaption02,
ShortAnsCaption03, ShortAnsCaption04)
VALUES
(0, 0, 1, 'A', '',
'', '', '', '', '', 0, 'GL', 0, '', 0,
'N', 'C', 'Report Date', '01NEW', '', '', '', '', '', '')
INSERT INTO RptFormat
(DisplayIndex, FileName, FormatName, ReportNbr)
VALUES
(0, '01NEWV1', 'Company ID', '01NEW'),
(1, '01NEWV2', 'Negatives In Parens', '01NEW')
```

Add the New Report to Existing Screens and Reports

To add the new report to the Report Control table:

1. Open *Screen Maintenance* (98.330.00).

Figure 28: Screen Maintenance (98.330.00)

2. In **Number**, type the number of your new report.
3. In **Name**, type the name you assigned to your report.
4. In **Module**, type the acronym for the module associated with your report, or press F3 and then select the module from the list.
5. In **Type**, select the appropriate option from the list. The options are:

- Report
 - Report with Interactive Process
 - SSRS Report
6. In **Menu Item**, Yes is already selected for you.
 7. Save and close the window.

Add or Modify a Group of Users of the Report

Go to “Give Users Rights to the New Menu Item” if you do not need to add or modify a group of users who will access your new report.

To add or modify a group:

1. Open *Group Maintenance (95.280.00)*.

Figure 29: Group Maintenance (95.280.00)

2. In **Group ID**, type or select a group identifier by pressing F3.
3. In **Name**, type a name for a new group. The name of an existing group appears automatically.
4. In **User ID**, type a user identifier to add a user to the group, or press F3 and select a user from the list.
5. For an existing group, review the users that are listed and make changes as needed.
6. (Optional) In **Home Page**, type the URL of the Web page that will appear when the user opens Microsoft Dynamics SL.
7. Save and close the screen.

Assign Users Access Rights to the New Report

In order for users other than administrators to access the new report, you must assign access rights to the report in *Access Rights Maintenance* (95.270.00).

To assign access rights to the new report:

1. Open Access Rights Maintenance (95.270.00).

Figure 30: Access Rights Maintenance (95.270.00)

2. Select Group from the **Type** list.
3. In **Group/User ID**, type the identifier of the group that will use your new report, or press F3 and select the group ID from the list.
The group name automatically appears.
4. Either select **All Companies**, or in **Company ID**, press F3 to select a company ID from the list.
5. Click **Preload**, and then select All or the appropriate module. Information about the screens and reports for the modules you selected appear in the grid.
6. In the detail grid, locate the new report, and assign the appropriate access rights for the group.
7. Save and close the screen.

Add the Report to a Menu

To add the report to a user menu:

1. Open *Menu Maintenance (98.350.00)*.

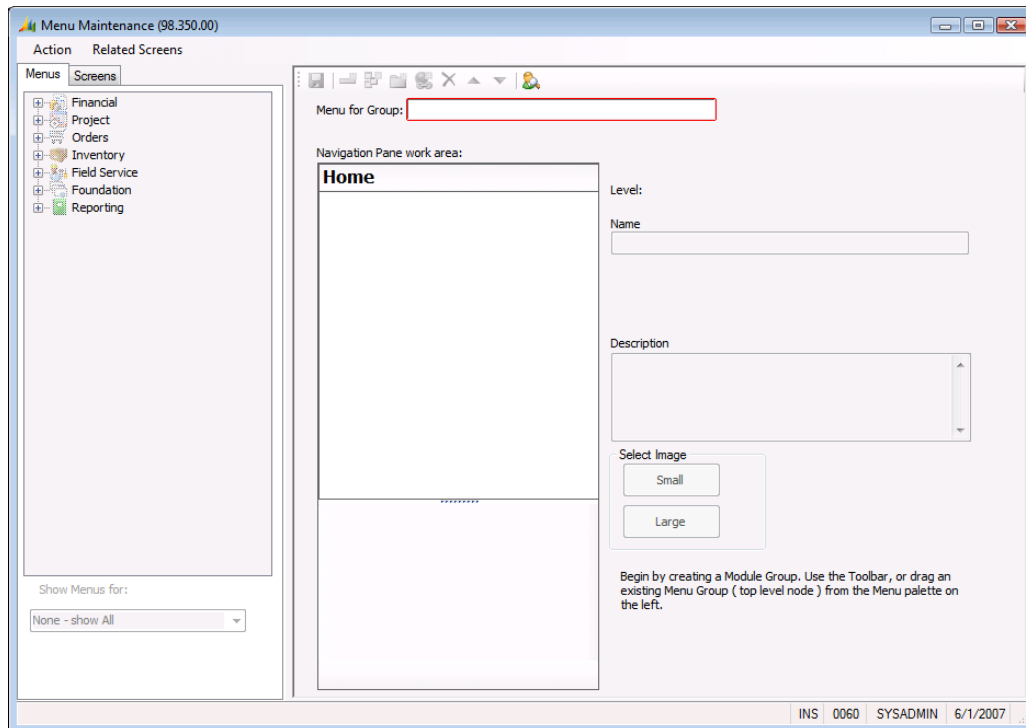





Figure 31: Menu Maintenance (98.350.00)

2. In **Menu for Group**, press **F3** and select the group that will access the new report.
3. If the group you selected has a custom menu system, it appears in the navigation pane work area. Go to Step 5.e below.

If the group does not have a custom menu system and information from the last *Menu Maintenance* session appears, use **Delete**  to clear the navigation pane work area.

Note: The software does not allow changes to the EVERYONE group's menu system.

4. For a new group, on the **Menus** tab, right-click the module group that is associated with the new report, and then click **Add Copy**. The module group's menus appear in the navigation pane work area.
5. In the navigation pane work area, expand the module that is associated with the new report.
6. Right-click the Report application group name, and then click **New Link** on the menu, or click the application group name and then click **New Link**  on the *Menu Maintenance* toolbar. A new untitled link is added at the bottom of the application group list.
7. In **Name**, type the name of the new report as you want it to appear on the menu.
8. In **Screen ID**, type the number of the report you are adding to the menu, or press **F3**, and select from the list.
9. A command line that will run the report appears automatically in **Command line for Application**. Make any changes that are needed.
10. In **Column**, select the column the report will appear in on the menu.
11. Save and then review your changes by clicking **Preview Menu** . The preview window appears.

12. In **Preview For User**, press **F3** and select a user who will access the new report.
13. In **Company**, you can select a company that is associated with the menu system or accept the default, and then press **TAB**.
The user's menu system appears.
14. Check to be sure the new report appears in the correct area on the appropriate module menu.
15. Close the preview window.
16. Close *Menu Maintenance* (98.350.00).

Adding a New Report Format

To add a new format to an existing report, you can use *Report Control Maintenance* (98.300.00) or the method outlined here. The example that follows uses a report that was created in Crystal Reports. The same concepts apply to SSRS reports.

This example explains how to add modified versions of the summary format of the *Trial Balance* (01.610.00) report to the **Report Format** selections in the General Ledger *Trial Balance* (01.610.00) ROI window.

To add a new format to an existing report:

1. Modify the 01610a report as described in “Creating a New Report from an Existing One with Crystal Reports.”
2. Save the file as 01610new.rpt in \Microsoft Dynamics\SL\Applications\Usr_Rpts.
3. Add the new report format to the **Report Format** list for the *Trial Balance* (01.610.00) report.

To do this, you can follow the steps listed in “Create a New Report Control Record” on page 55
OR

- a) open a database modification utility, log on to the appropriate system database, and run the following SQL statement:

```
SELECT * FROM RptFormat WHERE ReportNbr = '01610'
```

Note: In the case of the *Trial Balance* (01.610.00) report, there are two format records (Combined Totals and Debit and Credit Totals) with corresponding displayIndex of 0 and 1. The FormatName field stores the format name. Also note that the corresponding FileName field stores the report file name without the extension.

- b) Insert your own format in the unused format and name fields by running the following SQL statement:

```
INSERT INTO RptFormat
(DisplayIndex, FileName, FormatName, ReportNbr)
VALUES
(2, '01NEWV1', 'Company ID', '01610'),
(3, '01NEWV2', 'Negatives In Parens', '01610')
```

- c) Close the database utility and open the *Trial Balance* (01.610.00) report in Microsoft Dynamics SL.
- d) Open the Report Format list, and note the new format options.

Using SQL Expressions to Expand Reporting Capabilities

To create a report using SQL syntax not supported by the Crystal Reports standard record selection criteria, you can use the SQL Expressions feature. With this functionality, you can change your report's selection criteria by modifying the RI_WHERE field in the RptRuntime record. At runtime, the ROI reads the RptExtra records and then creates SQL expression fields that it uses with selection criteria you specify. This solution is available for use only in a pre-process, not through the ROI command line.

To use SQL Expression fields in your report:

1. Create an RptExtra database record that describes the SQL expression you want to use. Generate one record for each SQL expression you build, Be sure to populate the following fields:
 - RI_ID — Set to bRptRuntime.RI_ID.
 - DatabaseName — Set to bpes.DBName.
 - UserID — Set to bpes.UserID.
 - Parameters — Must start with "SQLExpression:", followed by the name of the SQL expression, an equal sign "=", and the expression. In the example below, the SQL expression is "SQL1".

Example:

```
Call SqlCursorEx(c_RptExtra, NOLEVEL, "c_RptExtra", "RptExtra", "")
sql(c_RptExtra, "Select * from RptExtra where RI_ID = " +
IParm(bRptRuntime.RI_ID)
bRptExtra.RI_ID = bRptRuntime.RI_ID
bRptExtra.parameters = "SQLExpression:SQL1=" + "DateDiff(Month,
IsNULL(PRBatInfo.WeekEnd, PRDoc.ChkDate), " + SParm(bRptRuntime.ReportDate)
+ ")"
bRptExtra.UserID = bpes.UserID
bRptExtra.DatabaseName = bpes.DBName
Call SInsert1(c_RptExtra, "RptExtra", bRptExtra)
```

2. Update the RI_WHERE field in the RptRuntime record to use the SQL expression. For the example in step 1, add the following to RI_WHERE:

```
{%SQL1} = 0
```

Hints and Tips

- It is a good idea to save your SQL statements in a script file so that they are readily available in the future. To do this, choose **File | Save As** from the utility's menu and enter a valid file name.
- Updates issued by Microsoft may change the same records addressed here. If such an event occurs, your SQL statement modifications will be lost. To recover, open the saved script file, modify the statements as necessary, and run them again.
- If you receive:
message 6203, "Report xxxxx not found in report library file"
- OR -
message 6202 "Unable to locate report library file xxxxxxx,"
this could indicate:
 - The report file name does not match the value in RptFormat.FormatName for that format.

Note: Report names are case sensitive, so make sure capital letters are used in all places.

 - The report number designated as the menu item does not match the value inserted into RptControl.ReportNbr for the report.

Note: Report numbers are also case sensitive, so make sure capital letters are used in all places.

 - The report number designated as the menu item and the value inserted into RptControl.ReportNbr for the report do not begin with the appropriate numeric prefix.
For example, an Accounts Payable report number (ReportNbr) must begin with the prefix 03. Similarly, an Accounts Receivable report number must begin with 08. These files should be located in \Microsoft Dynamics\SL\Applications\Usr_Rpts.

For more information, see "Maintaining Report Control Records" on page 25 or report control record information in the System Manager Help or user's guide.

Resolving Common Crystal Reports Setup Issues

The following topics discuss how to resolve issues that may arise when setting up Crystal Reports for access from a menu.

Missing User-Defined Function in Functions Pane

I do not see Microsoft Dynamics SL's user-defined functions in my *Functions* window.

Resolution:

Make sure U2LSol4.dll for Crystal Reports 2008 is in the \Program Files\Business Objects\BusinessObjects Enterprise 12.0\win32_x86 folder.

Incorrect Logon Parameters Error

An Incorrect Logon Parameters error appears when I attempt to print a report from within Microsoft Dynamics SL.

Resolutions:

- The ODBC package is not installed. Install the package.
- The ODBC package is not installed correctly. Reinstall the package.
- The Named Pipes protocol is not loaded. Named Pipes is the default protocol that is used by the SQL Server ODBC drivers. Add the Named Pipes protocol.

ODBC Error. Invalid Object Name

I want to modify a Microsoft Dynamics SL report, but when I attempt to preview the report using Crw32.exe outside of Microsoft Dynamics SL, I get an "ODBC Error. Invalid Object Name..." type of message, followed by a SQL Server error.

Resolution:

1. Choose **Database | Set Datasource Location** from the Crystal Reports menu. This opens the *Set Datasource Location* window.

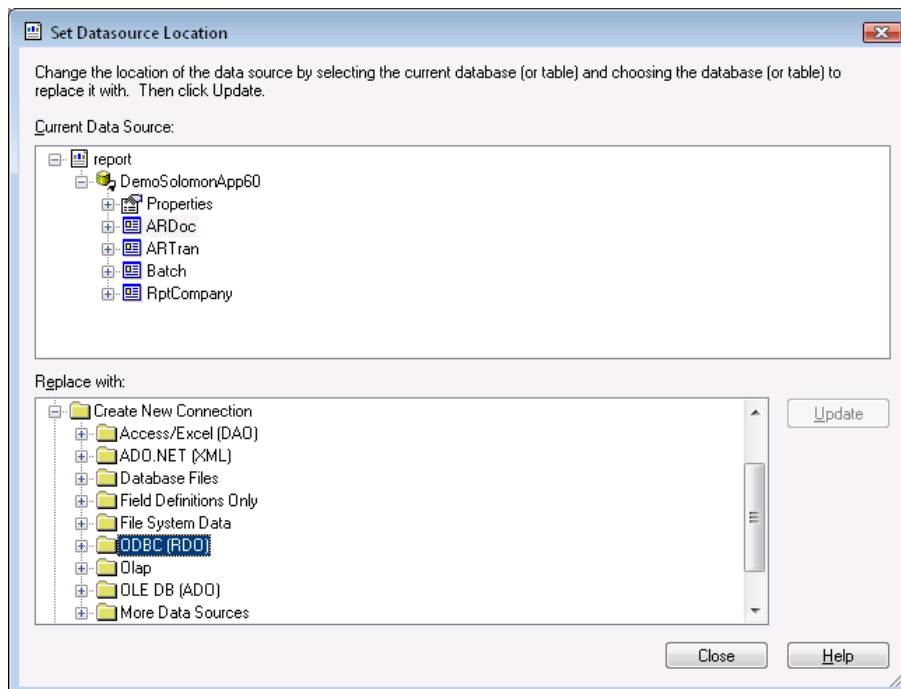


Figure 32: Set Datasource Location

2. Select a data source from the **Replace with** list.
3. Select the database table name.
The name you choose needs to match the name displayed in the **Table** field in the *Set Datasource Location* window.
4. Click **Update**.
5. Click **Close**.
6. Preview the report.

Note: This procedure is not required when printing reports through ROI. For databases you are logged on to, the software automatically performs this function.

Text Not Part of Formula Error

I want to modify a Microsoft Dynamics SL report, but when I attempt to preview the report using Crw32.exe, the message "The remaining text does not appear to be part of the formula" appears, and the text appearing is a user-defined function.

Resolution:

Crw32.exe cannot find the user-defined functions. See "Missing User-Defined Function in Functions ."

Purpose of User-Defined Functions

What is the purpose of user-defined functions?

Resolution:

User-defined functions have many purposes. They can be used to:

- Format data based on Flexdef definitions.
- Allow the reports access to module setup data without joining that table to the report.
- Format period numbers as they are in screens.
- Formatting address lines to not leave blank lines when data is incomplete.

How to Use User-Defined Functions

How can I use user-defined functions?

Resolution:

You can use user-defined functions in formulas created in a report. Follow these steps:

1. From the Crystal Reports menu, choose **Report | Formula Workshop**.
The *Formula Workshop* window appears.
2. Expand the **Formula Fields** folder, and right-click the name of the formula field you want to change.
3. Click **New**.
4. Enter the name of the new formula in **Formula Name**.
5. Click **Use Editor**. *Formula Editor* appears.

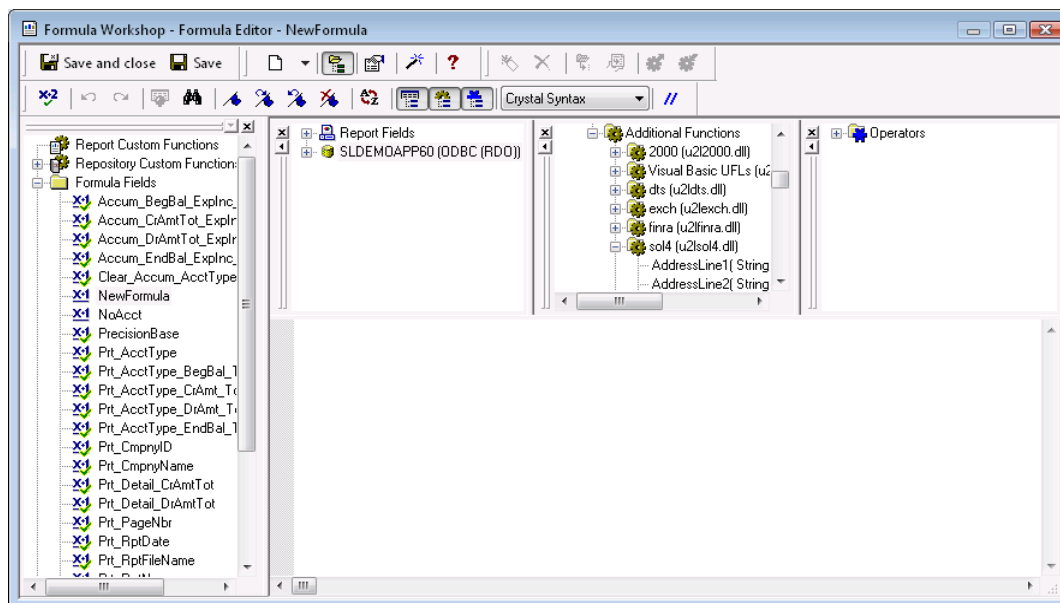


Figure 33: Formula Editor

The Microsoft Dynamics SL user-defined functions are listed in under **Additional Functions | Sol4** in the **Functions** pane. They begin with the AddressLine1. Choose a function as you would any other function in this dialog box.

If the user-defined functions do not appear on the list, you have an environment issue that needs to be resolved. See “Missing User-Defined Function in Functions Pane.”

Understanding Subreports

What Are Subreports?

Subreports are a Crystal Reports feature that allows reports (subreports) to be embedded within a parent report. Each subreport has its own SQL statement. This is beneficial if a report needs to obtain information just once during the report run. With subreports, your parent report's SQL statement does not require a Join to a database table you only need to access once.

Limits of Subreports

Each subreport is an additional report embedded within the parent report. Microsoft.Dynamics.SL.Reporting.dll modifies the parent report by cycling through the database tables and then changing the name of the database associated with the tables to be that of the current database. It performs the same sort of modification on subreports, and this modification occurs for each subreport within the parent report. While Microsoft.Dynamics.SL.Reporting.dll implements the code to modify a subreport, it does not modify the WHERE clause in subreport SQL statements.

Subreport Borders

Borders automatically appear around all subreports created. For consistency of report presentation, you may want to turn these borders off. Use Crystal Reports Designer to turn subreport borders off.

Understanding Drilldown Reports

What Are Drilldown Reports?

Drilldown reports are a Crystal Reports feature that displays an initial high-level presentation of report information with a capability that allows users to access the details that make up the high-level presentation (for example, see the details that make up period-to-date net sales totals). To drill down to the details of a report area, click the mouse in that area.

The top-level information of drilldown reports is viewable in the *Print Preview* window.

The Crystal Reports drilldown reporting feature is designed for distributed reporting situations.

Using Drilldown Reports

Drilldown reports will work with Microsoft Dynamics SL for displaying top-level report information in the *Print Preview* window. However, Microsoft Dynamics SL does not currently support drilling down into report data. You will not be able to view the detail portions of any drilldown reports. This is because the report details are suppressed when you create the drilldown reports and, therefore, unavailable for viewing.

RIParams

Report information parameters (RIParams) are report items defined by the ROI that control the type and length of the data printed on reports (for example, the report title and the number of characters in the report title). The following programs and files use RIParams when printing reports:

- The .rpt files
- Microsoft.Dynamics.SL.Reporting.dll
- CrpeSol4.dll
- U2LSol4.dll

Microsoft Dynamics SL reporting uses three types of RIParams: standard Microsoft Dynamics SL RIParams, user-inserted RIParams, and additional (custom) RIParams.

Note: All Microsoft Dynamics SL RIParams are display-only report items.

Microsoft Dynamics SL RIParams

The following table lists the RIParams used for standard Microsoft Dynamics SL reporting. ROI.exe automatically sets the value for each of these RIParams.

Name	Data Type * Length
AccessNbr	As String * 5
Acct	As String * 10
Banner	As String * 1
BatNbr	As String * 10
BegPerNbr	As String * 6
BusinessDate	As String * 10
CmpnyName	As String * 30
ComputerName	As String * 21
CpnyID	Char * 10
DatabaseName	As String * 20
DocNbr	As String * 10
EndPerNbr	As String * 6
LongAnswer(0 To 4)	As String * 80
NotesOn	As Integer
PerNbr	As String * 6
RepBegDate	As String * 10
RepEndDate	As String * 10
ReportDate	As String * 10
ReportFormat	As String * 30
ReportName	As String * 30
ReportNbr	As String * 5
ReportTitle	As String * 30
RI_BEGPAGE	As Integer
RI_CHKTIME	As String * 1
RI_COPIES	As Integer
RI_DATADIR	As String * 21
RI_DICTDIR	As String * 21
RI_DISABLED	As SmallInt

Name	Data Type * Length
RI_DISABLEQS	As SmallInt
RI_DISPERR	As String * 1
RI_ENDPAGE	As Integer
RI_ID	As Integer
RI_INCLUDE	As String * 1
RI_LIBRARY	As String * 80
RI_NOESC	As String * 1
RI_OUTAPPN	As String * 1
RI_OUTFILE	As String * 80
RI_PRINTER	As String * 1
RI_REPLACE	As String * 255
RI_REPORT	As String * 30
RI_STATUS	As String * 1
RI_TEST	As String * 1
RI_WHERE	As String * 255
RI_WPORT	As String * 40
RI_WPTR	As String * 40
RI_WTITLE	As String * 30
SegCustMask	As String * 16
SegCustTitle	As String * 16
SegInvenMask	As String * 26
SegInvenTitle	As String * 26
SegSubMask	As String * 38
SegSubTitle	As String * 38
SegVendMask	As String * 16
SegVendTitle	As String * 16
ShortAnswer(0 To 4)	As String * 10
Sub	As String * 24
SystemDate	As String * 10
SystemTime	As String * 7
UserId	As String * 47

Standard RParams

Standard RParams are pre-defined RParams that you can insert in a pre-process to control certain aspects of report printing. For example, you might want to insert the RParam RI_FONTITALIC to specify that the report should be printed in *italic* type style. Available RParams are listed below.

RParams could be individual fields from setup records or any other database records, or they could just be bits of information you want the report to be able to access.

RI_DEBUG

Determines if Microsoft.Dynamics.SL.Reporting.dll is in Debug mode.

RI_DRIVER

Identifies the name of the printer driver minus the .drv extension.

RI_DBTYPE

Identifies the database type against which to run the report. This is always set to MSSQL.

Value	Description
MSSQL	Microsoft SQL Server database
SSQL	Pervasive.SQL database

RI_FONTNAME

Identifies the font name used by the report, either an actual font name or the text {Report Font}. Microsoft.Dynamics.SL.Reporting.dll accesses this information and modifies the report at runtime. If {Report Font} appears in the RParam, no font modifications are performed on the current report.

RI_FONTSIZE

Identifies the font size used by the report. Microsoft Dynamics.SL.Reporting.dll accesses this information and modifies the report at runtime, dividing the font size value by 10 and using the result for report printing. This is acceptable because SWIM takes the value from the *Printer Options* (98.220.00) *FONT* window and multiplies this value by 10 before storing the result.

RI_FONTBOLD

Indicates whether or not the font identified by RI_FONTNAME should also be set in **boldface** type style.

Value	Description
0	Do not set the font to boldface
1	Set the font to boldface

Microsoft.Dynamics.SL.Reporting.dll accesses this information and modifies the report at runtime.

RI_FONTITALIC

Indicates whether or not the font identified by RI_FONTNAME should also be set in *italic* type style.

Value	Description
0	Do not set the font to italic
1	Set the font to italic

Microsoft.Dynamics.SL.Reporting.dll accesses this information and modifies the report at runtime.

RI_ORIENT

Indicates the printing orientation of the printer identified in *Printer Options (98.220.00)*.

Value	Description
1	Print pages in the portrait orientation
2	Print pages in the landscape orientation

Microsoft.Dynamics.SL.Reporting.dll accesses this information and modifies the report at runtime.

RI_USEPORIENT

Tells Microsoft.Dynamics.SL.Reporting.dll whether to print the report based on the report's orientation or the printer's orientation.

Value	Description
0	Use the report's orientation
-1	Use the printer's orientation

RI_FILETYPE

Identifies the type of file ROI creates when exporting a report. Microsoft.Dynamics.SL.Reporting.dll accesses this information and modifies the report at runtime.

Value	Description
0	Acrobat (PDF)
9	XML
A	Rich Text
B	Comma-Separated Values
C	Crystal Reports
M	HTML 3.2
L	HTML 4.0, / DHTML
T	Tab-Separated Values
W	Microsoft Office Word
X	Text
Y	Tab-Separated Text
5	Excel 97-2003
K	Excel 97-2003 (Data Only)

Creating New RIParams

When creating new RIParams, keep the following in mind:

- AddRIParam is an API provided in CrepSol4.dll that enables you to create RI_PARAMS. To use this function, you must first declare it as follows:

```
Declare Function AddRiParam Lib "CRPESOL4.DLL" (ByVal ri_id As Integer,  
ByVal ID As String, ByVal value As String) As Integer
```

- Microsoft Dynamics SL pre-processes can add parameters to the RI_PARAM table by calling the AddRiParam function, for example:

```
Function AddRiParam( RI_ID as integer, Name as string, value as string )  
as integer
```

- A return value of zero means everything is acceptable.
- A return value of 1 means you are over the 8 KB limit.
- A return value of 4 means the memory area is not set up.

User-defined Functions

Microsoft distributes user-defined functions as part of the Crystal Reports user function libraries, U2LSol4.dll and Crufladg.dll. Keep in mind that these functions are specific to Crystal Reports.

Note: For the Crystal Reports Designer (Crw32.exe) to use these functions, you should copy U2LSol4.dll to the \Program Files\Business Objects\BusinessObjects Enterprise 12.0\win32_x86 folder, and CrpeSol4.dll and Crufladg.dll to the \Windows\System32 folder.

U2LSol4.dll Functions

The user-defined functions described here deal with address functions. You can use these functions to eliminate the situation in which a printed address has a blank line because it does not contain all elements of a four-line address (for example, there is no second address line). In such a case, Addressline1 returns the first line of an address, Addressline2 returns the second address line, and the rest of the address follows this pattern. For example:

```
Addr1 =                1122 Crystal Ave.
Addr2 =
City =                 Fall City
State =                OH
Zip =                  33232
Country/Region =      USA
Addressline1 will return: 1122 Crystal Ave.
Addressline2 will return: Fall City, OH 33232
Addressline3 will return: USA
Addressline4 will return:
```

String AddressLine1(String, String, String, String, String, String)

Description

Obtain the first line of an address consisting of addr1, addr2, city, state, zip, country or region where any line may be completely empty.

Syntax

Any specific requirements for the input parameters—like the fact that FmtDate() only accepts date formats of mm/dd/yy or mm/dd/yyyy, or limits on the lengths of input strings—should be listed in this doc.

String AddressLine2(String, String, String, String, String, String)

Description

Obtain the second line of an address consisting of addr1, addr2, city, state, zip, country or region where any line may be completely empty.

Syntax

```
AddressLine2(addr1, addr2, city, state, zip, country/region)
```

String AddressLine3(String, String, String, String, String, String)

Description

Obtain the third line of an address consisting of addr1, addr2, city, state, zip, country or region where any line may be completely empty.

Syntax

```
AddressLine3(addr1, addr2, city, state, zip, country/region)
```


String AddressLine4(String, String, String, String, String, String)*Description*

Obtain the fourth line of an address consisting of addr1, addr2, city, state, zip, country or region where any line may be completely empty.

Syntax

```
AddressLine4(addr1, addr2, city, state, zip, country/region)
```

String APSetup(String)*Description*

Retrieve a column from the one row in the APSetup table. A snapshot of the Setup record is placed in memory by ROI.exe. This function reads from that memory to get the requested field value.

Syntax

```
APSetup(<Fieldname>)
```

Example

```
APSetup("CurrPerNbr")
```

This retrieves the APSetup.CurrPerNbr field from the memory area.

String CapFirst(String)*Description*

Capitalize the first letter of each word in a string.

Syntax

```
CapFirst(string)
```

String CDLL(DLLName, FunctionName, FunctionParameterString)*Description*

Call a specific routine in the specified .dll file.

Syntax

```
CDLL(DLLName, FunctionName, FunctionParameterString)
```

Function parameters are:

- DLLName – The name of the .dll file to load
- FunctionName – The name of the function to run
- FunctionParameterString – The list of the parameter to pass to the function, separated by spaces

String CmpnyName*Description*

Return the company name field from the RptRuntime record.

Syntax

```
CmpnyName
```

String CMSetup(String)*Description*

Retrieve a column from the one row in the CMSetup table. A snapshot of the setup record is placed in memory by ROI.exe. This function reads from that memory to get the requested field value.

Syntax

```
CMSetup (<Fieldname>)
```

Example

```
CMSetup ("CurrPerNbr")
```

This retrieves the CMSetup.CurrPerNbr field from the memory area.

Date FmtDate(String)*Description*

Accepts a date string of format *mm/dd/yy* or *mm/dd/yyyy* and converts the string to the Crystal Reports internal date format.

Syntax

```
FmtDate (date)
```

String FmtPhone(String)*Description*

Return the telephone number in the format (999)999-9999 Ext 99999.

Syntax

```
FmtPhone ({Vendor.phone})
```

String FmtScrnNbr(String)*Description*

Return the screen number (Microsoft Dynamics SL window number) in the format 99.999.

Syntax

```
FmtScrnNbr (Screen)
```

Example

```
FmtScrnNbr ("99999")
```

String FmtZip(String)*Description*

Return the zip code in the format 99999-9999.

Syntax

```
FmtZip ({Address.Zip})
```

String FRW1Per*Description*

Return a string in the format *For the Period XX Through YY* (XX is RptRuntime, Longanswer00; YY is RptRuntime, Longanswer01).

Syntax

```
FRW1Per ()
```

String FRW2Per*Description*

Return a string in the format *Period XX As of YY* (XX is RptRuntime, Longanswer02; YY is RptRuntime, Longanswer01).

Syntax

```
FRW2Per()
```

String FRW3Per*Description*

Return a string in the format *Fiscal Year XX* (XX is RptRuntime.begPernbr [last four digits]).

Syntax

```
FRW3Per()
```

String FSSetup(String)*Description*

Retrieve a column from the one row in the FSSetup table. A snapshot of the setup record is placed in memory by ROI.exe. This function reads from that memory to get the requested field value.

Syntax

```
FSSetup(<Fieldname>)
```

Example

```
FSSetup("CurrPerNbr")
```

This retrieves the FSSetup.CurrPerNbr field from the memory area.

String GLSetup(String)*Description*

Retrieve a column from the one row in the GLSetup table. A snapshot of the setup record is placed in memory by ROI.exe. This function reads from that memory to get the requested field value.

Syntax

```
GLSetup(<Fieldname>)
```

Example

```
GLSetup("NbrPer")
```

This retrieves the GLSetup.NbrPer field from the memory area.

String INSetup(String)*Description*

Retrieve a column from the one row in the INSetup table. A snapshot of the setup record is placed in memory by ROI.exe. This function reads from that memory to get the requested field value.

Syntax

```
INSetup(<Fieldname>)
```

Example

```
INSetup("CurrPerNbr")
```

This retrieves the INSetup.CurrPerNbr field from the memory area.

String Left(String, Number)

Description

Return the leftmost characters of a string. The number of returned characters is based on the length specified.

Syntax

```
Left(string, length)
```

String MapChar(String, Number, String, String)

Description

Check the source string to see if the character at the specified location is a certain value. If it is, then the char parameter is returned.

Syntax

```
MapChar(string, position, mapchar, tochar)
```

Example

```
MapChar ("1102324234", 5, "3", "TRUE")
```

This returns TRUE because the fifth character of 1102324234 is equal to 3.

String NameFlip(String)

Description

Returns a string similar to the source string except the contents are flipped around an embedded ~ character.

Syntax

```
NameFlip(name)
```

Example

```
NameFlip ("Bank of Baltimore~The")
```

This returns The Bank of Baltimore.

String PerAccessBusDate

Description

Returns the period number, the access number, and the business date (if the system date equals the business date). Uses the format:

```
(Period Number)-(Access Number) Bus: (Business Date)
```

With the Period Number in the format 99-99.

Syntax

```
PerAccessBusDate
```

String PerFmt(String)

Description

Returns the period in the format 99-99.

Syntax

```
PerFmt(period)
```

String POSetup(String)

Description

Retrieve a column from the one row in the POSetup table. A snapshot of the setup record is placed in memory by ROI.exe. This function reads from that memory to get the requested field value.

Syntax

```
POSetup(<Fieldname>)
```

Example

```
POSetup("CurrPerNbr")
```

This retrieves the `POSetup.CurrPerNbr` field from the memory area.

String ReptNumber*Description*

Returns the report number in the format *Rept 99.999*.

Syntax

```
ReptNumber
```

String Right(String, Number)*Description*

Return the rightmost characters of a string. The number of returned characters is based on the length specified.

Syntax

```
Right(string, length)
```

String RIPARAM(String)*Description*

Retrieve a value from the current RIPARAM memory area.

Syntax

```
RIPARAM(string)
```

Possible string value is:

RptNbr – Returns the Report Number

String RptDate*Description*

Returns the report date in the format:

Periods: (Beginning Period Number) Through (Ending Period Number) Period: (Beginning Period Number) As of: (Report Date)

With the beginning and ending period numbers in the format 99-99.

Syntax

```
RptDate
```

String RptName*Description*

Returns the name of the currently running report (this name is referred to off the menu).

Syntax

```
RptName
```

String SubStr2(String, Number)*Description*

Returns a substring of the specified string, starting at the position specified and continuing to the end of the string.

Syntax

```
SubStr(string, position)
```

Example

```
SubStr("Hi Bob, how are the kids", 8)
```

This returns *how are the kids*.

String SubStr(String, Number, Number)

Description

Returns a substring of the specified string, starting at the position specified and continuing to the end of the string.

Syntax

```
SubStr(string, position)
```

Example

```
SubStr("Hi Bob, how are the kids", 8, 3)
```

Returns *how*.

String SysDateTime

Description

Returns the computer systems time and date.

Syntax

```
SysDateTime
```

String Transform(SourceString, FormatString)

Description

Transform SourceString into a string that matches the format specified in the format string.

Syntax

```
Transform(expr, format)
```

Example

```
Transform("0000000000", "@RXX-XXX-XXX-XX")
```

This returns 00-000-000-00.

Special characters in the mask include:

- X – Letter
- x – Letter
- ! – Capitalize the letter

For all other characters, the value of the mask becomes part of the result.

String TransformNbr(Number, String)

Description

Transform the number into a string that matches the format specified in the format string.

Syntax

```
TransformNbr(Number, String)
```

Example

```
TransformNbr(45.99, "$99.9")
```

This returns \$46.0.

Special characters in the mask include:

- \$ – Shows up as is
- * – Shows up as is

- Others ignored
- 9 – number

String TransformNbrLocale ()

Description

Transform the number into a string using the localization options identified in FormatNumber.

Syntax

```
TransformNbrLocale (Number, FormatNumber)
```

Possible FormatNumber values are:

- 0 – Use current system locale
- 1 – Use locale decimal precision and decimal separator value
- 2 – Use locale thousands grouping and thousands separator value
- 4 – Format number as a negative value using system locale settings
- 16 – Use the currency settings for the current system locale

Possible values 1, 2, and 4 can be combined as in the following examples.

Example

```
TransformNumberLocal (12345.678, 0) returns 12,345.68 as the number format.
```

```
TransformNumberLocal (12345.678, 1) returns 12345.68 as the number format.
```

```
TransformNumberLocal (12345.678, 2) returns 12,346 as the number format.
```

```
TransformNumberLocal (12345.678, 1+2) returns 12,345.68 as the number format.
```

```
TransformNumberLocal (12345.678, 1+2+4) returns (12,345.68) as the number format.
```

```
TransformNumberLocal (12345.678, 16) returns $12,345.68 as the number format.
```

Crufladg.dll Functions

Following are function names and purposes in the Crufladg.dll.

BMSetup

Description

Retrieves data from the BMSetup table. The function takes in the field name of the BMSetup table and returns the data in that field. The function is used for Crystal Reports in order to incorporate table data without having to incorporate the table through linking.

Syntax

```
ADGclsADGCryFunctionsBMSetup(sField)
```

sField — The field name of the BMSetup table.

Example

```
ADGclsADGCryFunctionsBMSetup("DECPLHRS")
```

CheckDiffPOReqDetField

Description

Used in the 04.610 Change Order Report to determine if the current POReqDet field value equals the previous value.

Syntax

```
ADGclsADGCryFunctionsCheckDiffPOReqDetField(sField, sValue, sReqNbr,  
sReqCntr, sHeaderReqCntr, iLineNbr)
```

- sField — Field name being compared
- sValue — Value of the field being compared
- sReqNbr — ReqNbr of the PO
- sReqCntr — Revision number of the PO on the line level
- sHeaderReqCntr — Revision number of the PO on the header level
- iLineNbr — Line number

Returns True if the fields has changed from the last revision; false if it has not changed.

Example

```
ADGclsADGCryFunctionsCheckDiffPOReqDetField ("INVTID", {POReqDet.InvtID},  
{POReqDet.ReqNbr}, {POReqDet.ReqCntr}, {POReqHdr.ReqCntr},  
{POReqDet.LineNbr})
```

CheckDiffPOReqHdrField

Description

Used in the 04.610 Change Order Report to determine if the current POReqHdr field value equals the previous value.

Syntax

```
ADGclsADGCryFunctionsCheckDiffPOReqHdrField (sField, sValue, sReqNbr,  
sReqCntr)
```

- sField — Field name being compared
- sValue — Value of the field being compared
- sReqNbr — ReqNbr of the PO
- sReqCntr — Revision number of the PO on the line level

Returns True if the field has changed from the last revision; false if it has not changed.

Example


```
ADGclsADGCryFunctionsCheckDiffPOReqHdrField ("SHIPVIA",
Trim({POReqHdr.ShipVia}), {POReqHdr.ReqNbr}, {POReqHdr.ReqCntr})
```

ConvertAmtFromStdUnits

Description

Interface to call the ADGUnitConversion.ConvAmtFromStdUnits function used to convert a monetary amount from standard units. (Not Used)

Syntax

```
ADGclsADGCryFunctionsConvertAmtFromStdUnits(Amt, ConvFactor, UnitMultDiv,
DecimalPlaces)
```

- Amt – Amount (in standard units) to be converted
- ConvFactor – Conversion factor
- UnitMultDiv – Multiply/divide indicator (M/D)
- DecimalPlaces – Number of decimal places to round to. If not specified, no rounding is performed

Returns the converted amount.

Example

```
ADGclsADGCryFunctionsConvertAmtFromStdUnits({SOLine.QtyOrd},
{SOLine.CnvFact}, {SOLine {SOLine.UnitMultDiv}, {INSetup.DecPlQty}}
```

ConvertAmtToStdUnits

Description

Public interface to call the ADGUnitConversion.ConvAmtToStdUnits function used to convert a monetary amount to standard units. (Used in 40.790 Early Warning Report)

Syntax

```
ADGclsADGCryFunctionsConvertAmtToStdUnits(Amt, ConvFactor, UnitMultDiv,
DecimalPlaces)
```

- Amt – Amount (in non-standard units) to be converted
- ConvFactor – Conversion factor
- UnitMultDiv – Multiply/divide indicator (M/D)
- DecimalPlaces – Number of decimal places to round to. If not specified, no rounding is performed

Returns the converted amount.

Example

```
ADGclsADGCryFunctionsConvertAmtToStdUnits({SOLine.QtyOrd}, {SOLine.CnvFact},
{SOLine {SOLine.UnitMultDiv}, {INSetup.DecPlQty}}
```

FormatChainDiscount

Description

Formats an unformatted chain discount using the clsADGChainDisc. (Used in OM form reports.)

Syntax

```
ADGclsADGCryFunctionsFormatChainDiscount(sSubAcct)
```

sSubAcct – Subaccount formatted number

Returns the subaccount description.

Example

```
ADGclsADGCryFunctionsFormatChainDiscount({SOLine.APClearingSub})
```

FormatPrice

Description

This is a function that will override the built-in formatting of numbers in Crystal Reports. (Not used)

Syntax

```
ADGclsADGCryFunctionsFormatPrice(sDecimalPlaces, sCuryID)
```

- **sDecimalPlaces** String The number of decimal places as specified by the XXSetup table.
- **Optional sCuryID** String The currency ID from the main table used in the report. This will be used in later versions of the function.

Example

```
ADGclsADGCryFunctionsFormatPrice({INSetup.DecPlPrcCst}, {PurchOrd.CuryID})
```

GetAccountDescr

Description

Gets the accounts description. This is used to get descriptions for accounts that cannot be joined in a report like INSetup, POSetup accts, etc.

Syntax

```
ADGclsADGCryFunctionsGetAccountDesc(sAcct)
```

sAcct – Account number

Returns the account description.

Example

```
ADGclsADGCryFunctionsGetAccountDesc({INSetup.APClearingAcct})
```

GetCurrentInvoiceNbr

Description

Public interface for Crystal Reports to get the current invoice number in for pre-printed invoices and write that number to SOPrintQueue. (Used in Invoice rpt)

Syntax

```
ADGclsADGCryFunctionsGetCurrentInvoiceNbr(CpnyID, ReportName)
```

- **CpnyID** – CpnyID field from SOShipHeader
- **ReportName** – Current report format

Returns the next invoice number from SOType.

Example

```
ADGclsADGCryFunctionsGetCurrentInvoiceNbr ((trim(riparam("CpnyId"))),  
trim(riparam("ReportName")))
```

GetNextInvoiceNumber

Description

Public interface for Crystal Reports to get the next invoice number in for pre-printed invoices and write that number to SOPrintQueue. (Used in Invoice rpt)

Syntax

```
ADGclsADGCryFunctionsGetNextInvoiceNumber(CpnyID, RI_ID, ShipperID,  
ReportName)
```

- **CpnyID** –CpnyID field from SOShipHeader
- **RI_ID** – Current RI_ID for the print job from SOPrintQueue
- **ShipperID** – ShipperID from SOShipHeader
- **ReportName** – Current report format

Returns the next invoice number from SOType.

Example

```
ADGclsADGCryFunctionsGetNextInvoiceNumber ({SOShipHeader.CpnyID},
{SOPrintQueue.RI_ID}, {SOPrintQueue.ShipperID}, trim(riparam("ReportName")),
PageNumber)
```

GetPeriodFromDate

Description

Public interface for Crystal Reports to call the GetPeriodFromDate function in the ADGPeriod module. This function takes a given date and gets the period for it.

Syntax

```
ADGclsADGCryFunctionsGetPeriodFromDate (PerDate)
```

PerDate – Date to find the period for

Returns the period where the passed in date falls.

Example

```
ADGclsADGCryFunctionsGetPeriodFromDate ({INTran.PerPost})
```

GetQtyAvail

Description

Public interface for Crystal Reports to call the clsADGPlan.GetQtyAvail function used to get the quantity available as of a specified date. (Used in 40.790 Early Warning Report)

Syntax

```
ADGclsADGCryFunctionsGetQtyAvail (InvtID, SiteID, AvailDateVB)
```

- InvtID – Inventory ID
- SiteID – Site ID
- AvailDateVB – Date to check availability for

Returns the quantity available as of the end of the day on the specified date.

Example

```
ADGclsADGCryFunctionsGetQtyAvail ({SOLine.InvtID}, {SOLine.SiteID}, Today)
```

GetQtyAvailable

Description

Gets the Net Quantity Available up to the passed Lead Time Date.

Syntax

```
ADGclsADGCryFunctionsGetQtyAvailable (InvtID, SiteID, LeadTimeDate)
```

- InvtID – Inventory ID
- SiteID – Site ID
- LeadTimeDate – Date through which to check (string, mm/dd/yyyy)

Returns the Net Qty Available – On Hand Qty + Supply/Demand Qtys.

Example

```
ADGclsADGCryFunctionsGetQtyAvailable ({SOLine.InvtID}, {SOLine.SiteID},
Today)
```

GetQtyAvailEx

Description

Public interface for Crystal Reports to call the `clsADGPlan.GetQtyAvail` function used to get the quantity available as of a specified date. This is the same as `GetQtyAvail`, except it has the `UseInfLeadTimeQty` parm.

Syntax

```
ADGclsADGCryFunctionsGetQtyAvailEx(InvtID, SiteID, AvailDateVB,  
UseInfLeadTimeQty)
```

- InvtID – Inventory ID
- SiteID – Site ID
- AvailDateVB – Date to check availability for
- UseInfLeadTimeQty – Flag to specify whether to consider lead time availability as infinity or not (IR support)

Returns the quantity available as of the end of the day on the specified date.

Example

```
ADGclsADGCryFunctionsGetQtyAvailEx ({SOLine.InvtID}, {SOLine.SiteID}, Today,  
True)
```

GetQtyAvailToday

Description

Public interface for Crystal Reports to call the `clsADGPlan.GetQtyAvailToday` function used to get the quantity available today. (Used in the Order Management *Early Warning Report* (40.790.00).)

Syntax

```
ADGclsADGCryFunctionsGetQtyAvailToday (InvtID, SiteID)
```

- InvtID – Inventory ID
- SiteID – Site ID

Returns the quantity available as of the end of the day on the specified date.

Example

```
ADGclsADGCryFunctionsGetQtyAvailToday ({SOLine.InvtID}, {SOLine.SiteID})
```

GetScreenName

Description

Gets the screen name from the Screen table for a given screen number.

Syntax

```
ADGclsADGCryFunctionsGetScreenName (ScreenNbr)
```

ScreenNbr – Screen number field

Returns the next invoice number from `SOType`.

Example

```
ADGclsADGCryFunctionsGetScreenName ({INTran.Crtd_Prog})
```

GetSubAccount

Description

Gets the subaccounts description. This function takes a subaccount, strips out all of the formatting, and grabs the subaccounts description.

Syntax

```
ADGclsADGCryFunctionsGetSubAcct (sSubAcct)
sSubAcct – Subaccount formatted number
```

Returns the subaccount description.

Example

```
ADGclsADGCryFunctionsGetSubAcct ({INTran.COGSSub})
```

GetTotal_Item2Hist_PTDQtySls

Description

Public interface for Crystal Reports to get the Total Qty Sold from Item2Hist over a period of time defined by the current period minus the number of periods.

Syntax

```
ADGclsADGCryFunctionsGetTotalItem2HistPTDQtySls (InvtID, SiteID, Period,
NumberOfPeriods)
```

- InvtID – Inventory ID
- SiteID – Site ID
- Period – Period at which the user would like to set as the current period
- NumberOfPeriods – Number of periods back in history that the user would like to accumulate

Returns the Total Qty Sold over a given number of periods.

Example

```
ADGclsADGCryFunctionsGetTotalItem2HistPTDQtySls ({ItemSite.InvtID},
{ItemSite.SiteID}, {@ReportPeriod}, Val(GLSetup( "NbrPer" )))
```

GetTotal_ItemHist_PTDCOGS

Description

Public interface for Crystal Reports to get the Total Cost of Goods Sold from ItemHist over a period of time defined by the current period minus the number of periods.

Syntax

```
ADGclsADGCryFunctionsGetTotalItemHistPTDCOGS (InvtID, SiteID, Period,
NumberOfPeriods)
```

- InvtID – Inventory ID
- SiteID – Site ID
- Period – Period at which the user would like to set as the current period
- NumberOfPeriods – Number of historical periods to query

Returns the Total Cost Of Goods Sold over a given number of periods.

Example

```
ADGclsADGCryFunctionsGetTotalItemHistPTDCOGS ({ItemSite.InvtID},
{ItemSite.SiteID}, {@ReportPeriod}, Val(GLSetup( "NbrPer" )))
```

GetTotal_ItemHist_PTDCostRcvd

Description

Public interface for Crystal Reports to get the Total Cost Received from ItemHist over a period of time defined by the current period minus the number of periods.

Syntax

```
ADGclsADGCryFunctionsGetTotalItemHistPTDCostRcvd (InvtID, SiteID, Period,  
NumberOfPeriods)
```

- InvtID – Inventory ID
- SiteID – Site ID
- Period – Period at which the user would like to set as the current period
- NumberOfPeriods – Number of historical periods to query

Returns the Total Cost Received over a given number of periods.

Example

```
ADGclsADGCryFunctionsGetTotalItemHistPTDCostRcvd ({ItemSite.InvtID},  
{ItemSite.SiteID}, {@ReportPeriod}, Val(GLSetup( "NbrPer" )))
```

GetTotal_ItemHist_PTDSIs

Description

Public interface for Crystal Reports to get the Total Sales from ItemHist over a period of time defined by the current period minus the number of periods.

Syntax

```
ADGclsADGCryFunctionsGetTotalItemHistPTDSIs (InvtID, SiteID, Period,  
NumberOfPeriods)
```

- InvtID – Inventory ID
- SiteID – Site ID
- Period – Period at which the user would like to set as the current period
- NumberOfPeriods – Number of historical periods to query.

Returns the Total Qty Sold over a given number of periods.

Example

```
ADGclsADGCryFunctionsGetTotalItemHistPTDSIs ({ItemSite.InvtID},  
{ItemSite.SiteID}, {@ReportPeriod}, Val(GLSetup( "NbrPer" )))
```

INSetup

Description

Retrieves data from the INSetup table. The function takes in the field name of the INSetup table and returns the data in that field. The function is used for Crystal Reports in order to incorporate this table data without having to incorporate the table through linking.

Syntax

```
ADGclsADGCryFunctionsINSetup(sField)
```

sField – Field name of the INSetup table

Returns the data from the INSetup table for that field.

Example

```
ADGclsADGCryFunctionsINSetup("AllowCostEntry")
```

IRSetup

Description

Retrieves data from the IRSetup table. The function takes in the field name of the IRSetup table and returns the data in that field. The function is used for Crystal Reports in order to incorporate a table's data without having to incorporate the table through linking.

Syntax

```
ADGclsADGCryFunctionsIRSetup(sField)
```

sField – Field name of the IRSetup table

Returns the data from the IRSetup table for that field.

Example

```
ADGclsADGCryFunctionsIRSetup("DemPeriodNbr")
```

IsCurrentPOReqDetLine

Description

Check to see if this is the current detail record for this PO revision. This is a fix for the problem of printing POs with site detail in site order for 04.600 and 04.610.

Syntax

```
ADGclsADGCryFunctionsIsCurrentPOReqDetLine (HeaderReqNbr, HeaderReqCntr,
DetLineNbr, DetReqCntr)
```

- HeaderReqNbr –ReqNbr on the Header level
- HeaderReqCntr – Revision number on the header level
- DetLineNbr –Detail line number
- DetReqCntr – Revision number on the detail level

Returns true if this is the latest version.

Example

```
ADGclsADGCryFunctionsIsCurrentPOReqDetLine ({POReqHdr.ReqNbr},
{POReqHdr.ReqCntr}, {POReqDet.LineNbr}, {POReqDet.ReqCntr})
```

PeriodPlusPeriodNum

Description

Public interface for Crystal Reports to call the PeriodPlusPeriodNum. This function takes a period and adds the number of days.

Syntax

```
ADGclsADGCryFunctionsPeriodPlusPeriodNum(Period, Number)
```

- Period – Period number to be added to
- Number – Number of periods to be added

Returns the new period.

Example

```
ADGclsADGCryFunctionsPeriodPlusPeriodNum({INTran.PerNbr}, 5)
```

POSetup

Description

Retrieves data from the POSetup table.

Syntax

```
ADGclsADGCryFunctionsPOSetup(sField)
```

sField — Field name of the POSetup table

Returns the data from the POSetup table for that field.

Example

```
ADGclsADGCryFunctionsPOSetup("AllowCostEntry")
```

Microsoft Dynamics SLStartup

Description

Initializes the tables and calls the ApplInit procedure for some of the Crystal Reports functions.

SOSetup

Description

Retrieves data from the SOSetup table.

Syntax

```
ADGclsADGCryFunctionsSOSetup(sField)
```

sField — Field name of the SOSetup table

Returns the data from the SOSetup table for that field.

Example

```
ADGclsADGCryFunctionsSOSetup("AllowCostEntry")
```

WOSetup

Description

Retrieves data from the WOSetup table. The function takes in the field name of the WOSetup table and returns the data in that field. The function is used for Crystal Reports in order to incorporate a table's data without having to incorporate the table through linking.

Syntax

```
ADGclsADGCryFunctionsWOSetup(sField)
```

sField — Field name of the WOSetup table

Returns the data from the WOSetup table for the field.

Example

```
ADGclsADGCryFunctionsWOSetup("AllowCostEntry")
```


Adding SNote Fields to Crystal Reports

The following topics explain how to add SNote fields to a standard report written with Crystal Reports.

Background Information

RptRuntime.NotesOn is a field in the RptRuntime Table that instructs ROI.exe to print the note fields which are placed on the report. The report that is created or modified uses Crystal Reports subreport technology. The elements that you will work with are:

- Conditional Sections – Allow report designers to create sections of a report that can be printed based on some sort of condition or setting.
- Subreport – Allow the creation of reports called from a main report to print. These reports have their own SQL query statement associated with them.

Steps to Add SNote Fields

To create a new section on a report, and add an SNote Fields subreport to that section:

1. Use Crystal Reports to open the report you want to modify.
2. Select Report | Section Expert from the menu.
3. Choose the report section where you want to add the new section.
4. Click **Insert** on the *Section Expert* window.
5. Make sure the new section is highlighted in the **Sections** list.
6. Press the **Formula** button to the right of **Suppress** (no drill-down).



7. Set the following text for the Suppress formula in the *Format Formula Editor* window:

```
if RIPARAM("NOTESON") = "1" then
FALSE
else
TRUE
```

8. Click **Check**  or press ALT+C.

This will inform you of any errors that need to be addressed at this point.


9. Click **Save and Close**.


Notice the **Formula** button has changed to denote the section is now a conditional section. 

10. Click **OK** to return to the Report Designer.
11. Make the new section visible so you can add the subreport.
12. Choose **Insert | Subreport** from the menu.
13. Click **Create a subreport with the Report Wizard**.
14. Type a report name (for example, SNote Fields).
15. Click **Report Wizard**.
16. If you have logged on to a data source:
 - a) In the *Report Wizard* window under **Available Data Sources**, expand the **Current Connections** folder
 - b) Select an appropriate ODBC data source.

If you have not logged on to a data source:

 - a) Expand Create New Connection | ODBC (RDO), and then double-click Make New Connection.
 - b) Select a database name, click Next.

- c) Log on using your SQL Server SA password.
 - d) Click Finish.
17. Locate SNote in the list of database tables, and double-click to add it to the **Selected Tables** list.
 18. Click **OK**, and then click **Next**.
 19. On the Report Wizard window, double-click **SNoteText** or click to select it and then click  to add it to the **Fields to Display** list.
 20. Click **Finish** to return to *Insert Subreport*.
 21. Click the **Link** tab.
 22. From the **Available Fields** list under **Container Report field(s) to link to**, locate a Noteid field for the SNote records you want to show on the report.

This could be a field like **Vendor.Noteid** if you want to print a vendor's SNote records.
 23. Double-click the Noteid field or click to select it and then click  to add it to the **Field(s) to link to** list.
 24. Verify that the field name under the **Select data in subreport based on field** check box is correct.
- Note:** The field name should end with "nID". If you are unable to select the SNotes.nID field, perform these steps again to correct the problem.
25. Click **OK** in the *Insert Subreport* window.
 26. Return to the *Report Design* window, and find the box representing the subreport you just created.

Use the cursor to locate a transparent box with a light border. The report name you selected appears in the box.
 27. Drop this box into the new conditional section. Notice that a new tab appears at the top of the *Report Design* window.

The tab name is the name of the subreport you created.
 28. Click the subreport tab and format your subreport as needed.

Notice that Crystal Reports put a border around the subreport. You can leave it or remove it.
You may want to suppress the report headers and footers.

Note: Keep in mind that the SNote fields will now print on the report only when Print Notes on the ROI window is selected.

Getting More Crystal Reports Information

You can find more information about using Crystal Reports with Microsoft Dynamics SL by accessing the Knowledge Base at

mbs.microsoft.com/customersource/northamerica/search/pages/resultskb.aspx or find general information about Crystal Reports by accessing sap.com/pc/analytics/business-intelligence/software/crystal-reports/index.html. Use the Crystal Reports Help or user's guide for details on changing or creating reports.

Using SQL Server Reporting Services

When you create a report using SQL Server Reporting Services (SSRS), there are a few procedures that you will need to carry out to take advantage of the features of the Report Option Interpreter (ROISRS). These requirements, as well as ROISRS features, are explained in the topics that follow.

Note: The names you give your SSRS report files must have .rdl file extensions so that the ROISRS can locate them. Do not use the .rdlc file extension.

ROISRS Features

Dynamic Record Selection

The ROISRS offers the ability to change the selection criteria for a report at runtime. For this feature to work properly, the ROISRS must recognize the data source to which the selection change will be applied. You can define more than one data source for a SSRS report. However, the ROISRS needs to know which data source to modify. The following rules apply:

- If only one report data source is defined, name the data source SLReportDataSource. The ROISRS will modify its statement based on the user's input.
- If more than one report data source is defined, the ROISRS expects one of the data sources to be named SLReportDataSource. When that data source is found, the ROISRS will modify the statement based on user input and the contents of the ROISRS screen. If the ROISRS does not find a data source named SLReportDataSource, it will use the first data source in the report and modify its statement based on the user's input.
- It is recommended that you do **not** use shared data sources, but instead, define new data sources for each report. A data source identifies the tables that are accessed to create a report. If a data source is shared between two different reports, one of the reports might access data it cannot use. This can be costly in terms of performance. However, it may be beneficial to share a data source between different formats of the same report.

These same requirements apply if your report uses subreports.

If you name your data source correctly, the ROISRS will be able to modify the reports Select statement for the following:

- **Select** tab

Note: When you press F3 under **Field** on the **Select** tab, the report is accessed to determine which fields are available. The list of fields includes any that are included in the report's Query statement. If you would like to reduce the size of this list, implement the ROISelect feature (for more information, see "**ROISelect**").

- Period fields
- Company selections

ROISelect

The ROISelect feature helps the report designer limit the choices a user has in the **Sort** and **Select** tabs of the ROI and ROISRS screen. By default, the possible values for **Field** on these tabs include every field returned by the report's main data source. To modify this list, you can create a function in the report's code section that defines the information which will appear in the possible values list. See "How to Use ROISelect" on page 50 for more details.

Print Preview

Review a report before printing.

Print

Print a report. You can select such options as the number of copies to print, a page range, and the printer you will use.

Export

Export the contents of the report, including formatting, to a file in several different file formats.

Note: If you are writing a report for the specific purpose of exporting the data to an Excel spreadsheet, we recommend that you design the report to use tables. The table format will clearly define the columns the data is in, much like a spreadsheet.

Change a Report Font

By modifying settings in *Printer Options* (98.220.00), you can change the font of the report at runtime. Every character on the report will print in the font you select. No special settings are needed in your report in order for the ROI to recognize the font change.

Print a Report Cover Page

On the **Cover Page** tab, select **Print Cover Page** and type a description.

Embedding Custom Code

You can embed custom code in your SSRS reports. The ROISRS will modify this code at runtime to add functionality, but it will not permanently change the code you create.

Dynamic Sorting

When you create a report, the data is sorted based on the Order By clause in the report's SQL query statement. Sort order can be ascending or descending. The field names and their sort direction appear on the **Sort** tab. These items are marked as *Sort Type* or *Sort Field*.

If you add an Order By clause to your report's Query statement, the items will appear. If a user adds or modifies the contents of the **Sort** tab, the ROISRS will update the report before displaying it.

Dynamic Report Groupings

You can implement dynamic grouping in a report by using tables, lists, matrices, or rectangles. If you use both or more than one set of either in the same report, dynamic grouping will not be available. You will **not** see the group definitions on the ROISRS **Sort** tab. Also, the ROISRS only supports the modification of table and list groupings. Page break settings are available.

Dynamic Notes Printing

The ROISRS provides a check box for reports that contain notes (the Snote.NoteText field). If the check box is selected, the value of *noteson* in the report information parameters (RIParams) is set to 1 (see

the example below) so that SNote data will appear on the report. If the expression returns False, the note (Textbox, in this case) will be visible.

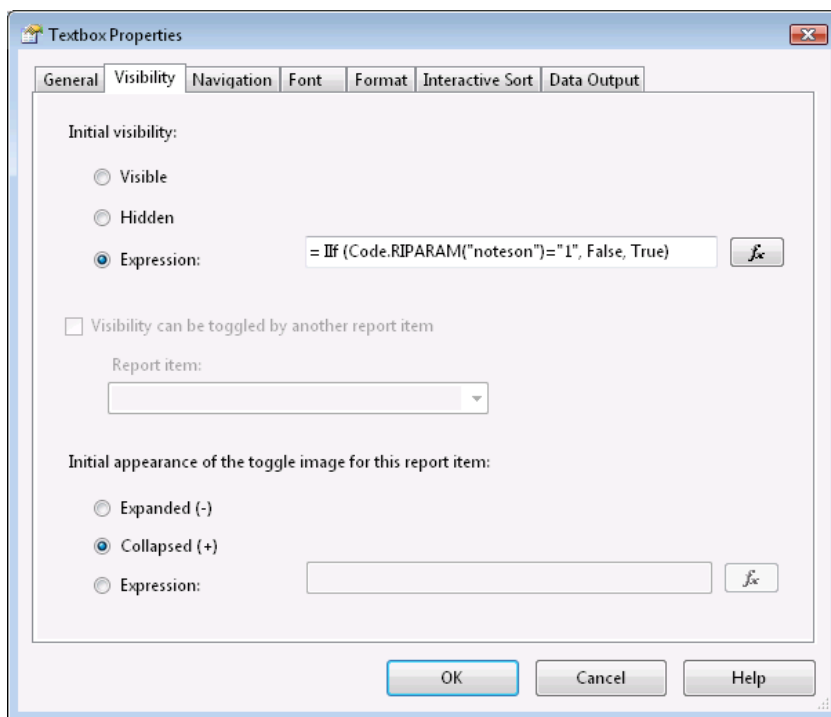


Figure 34: Visual Studio Textbox Properties, Visibility tab

When you design the report, you will need to create a RIParam function in the code section. This will allow you to view temporary data when you preview your report inside the Visual Studio IDE.

Sample code is provided to assist you. The sample, ReportCode.txt, is located in the Usr_Rpts folder of your Microsoft Dynamics SL installation. At runtime, the ROI will recreate the RIParam function in the code section so that “real” data appears on your reports.

Note: You could also show SNote data using a subreport that links its data source to the main report.

Report Templates

No special report setup is required.

Multiple Formats for a Single Report

Define your report in the *Report Control Maintenance* (98.300.00) screen, and specify different report names for each format of the same report.

Custom Report Storage

When the ROISRS loads a report, it follows the same logic as the ROI. It looks first in the `\Microsoft Dynamics\SL\Applications\Usr_Rpts` folder. If the report file is not found, the folder of the module associated with the report is searched. For example if it is attempting to load the General Ledger report 01650.rdl, the ROISRS will look in the Usr_Rpts folder first, and then in the GL folder.

Passing Data to a Report

If your report needs information passed from a pre-process, the RIParam function can provide the solution. The ROISRS creates several pieces of information that can be used or displayed on the report. At runtime, it will recreate the RIParam function in the report’s code section. All RIParams defined by the ROISRS and a pre-process are handled by the RIParam function. If you would like to use the RIParam data, use the expression field on a text box to specify which piece of data you want to

use. For example, entering `Code.RIPARAM("Accessnbr")` in the **Value** field of the field properties would cause the current access number to print on the report. The figure below illustrates how to use an expression if you want to display the current access number:

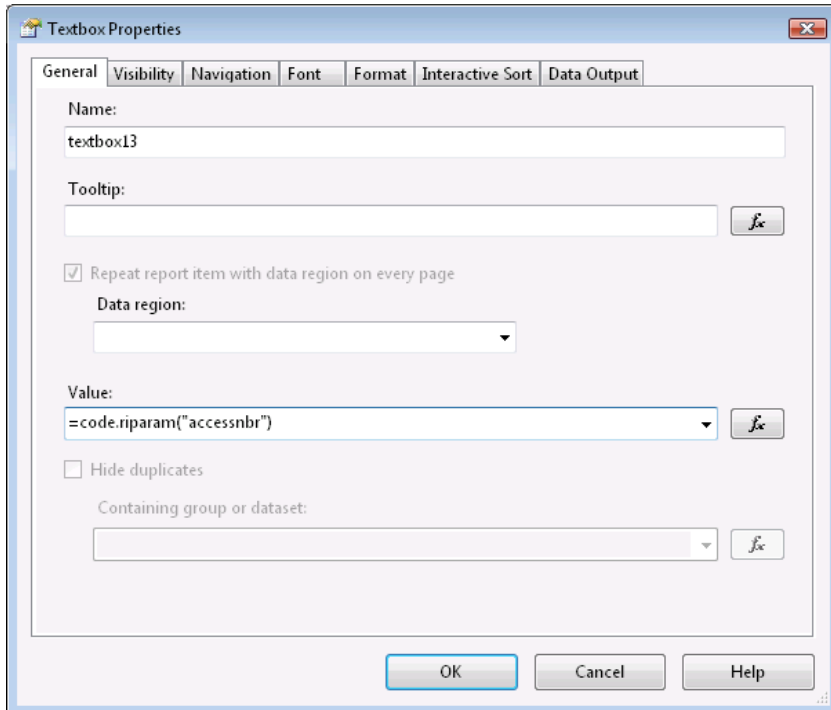


Figure 35: Visual Studio Textbox Properties, General tab

Since you have referenced the code section of the report in an expression, your report will not build unless the code exists. Create a stub RIPARAM function for use in testing and previewing the report. At runtime, the ROISRS will replace your stub function with a working RIPARAM function that has accurate values. For example:

```
Function RIPARAM( Byval ID as string ) as string
    RIPARAM = "Test Data"
End Function
```

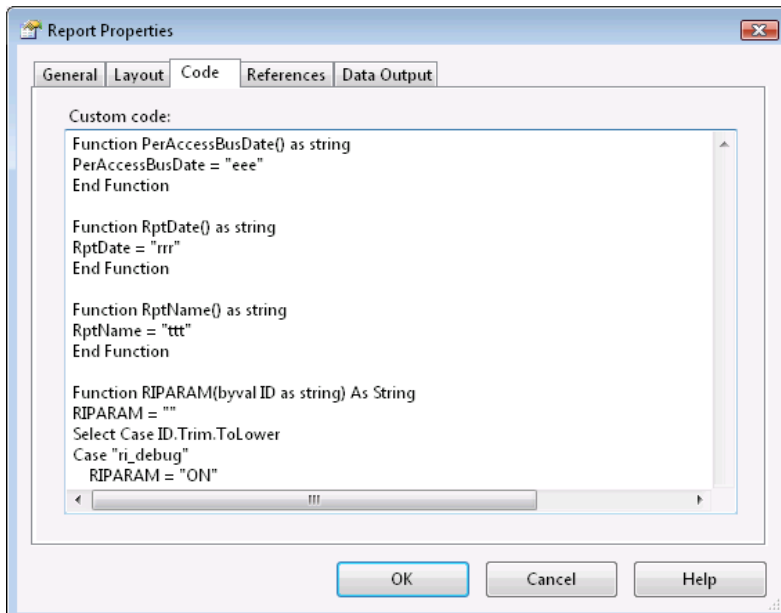


Figure 36: Visual Studio Report Properties, Code tab

Calling Pre-processes and Post-processes

Processes created for existing Crystal reports can be used with SSRS reports as well.

Subreport

A subreport can be a useful tool for presenting data. If you choose different selection criteria for your main report, you can make the same modification in your subreport's data source by telling the ROISRS to modify the subreport query statement in the same manner as that of the main report. To do this, create a function in the subreport's code section named `ModifySubReportsql`, and set its return value to `True`.

```
Function ModifySubReportsql() as String
    ModifySubReportsql = "TRUE"
End Function
```

If the function is present in the subreport, the subreport's data sources will be modified. If there is more than one data source in the subreport, query statements for all data sources with names that start with `SLReport` are modified.

Subreports are stored in their own file in the same location as the main report. If you modify a standard report and move it to the `Usr_Rpts` folder, you must also move the subreport file that it uses to the same folder.

Note: The ROISRS does not support the modification of subreports that also have subreports.

SQL Query as Text

You can use the `SL_Query()` function to view the SQL call that produces a report. To do this, create a stub function in the report's code section.

```
Function SL_Query() as string
    SL_Query = "hi"
End Function
```

At runtime, the ROISRS will replace `hi` with the actual SQL query that produces the report. You can also set a text box expression to `=Code.SL_Query()` to display the value on the report.

Creating a New Report from an Existing One Using SSRS

The example below demonstrates how to create a new SQL Server Reporting Services (SSRS) report by modifying an existing one. This example uses the combined totals format of the *Trial Balance (SSRS) (01.611.00)* report.

To create a new report using SSRS:

1. Open Visual Studio 2005.
2. On the File menu, click **Open**.
3. If you have already created a report project, you can use that project for this work. Go to Step 7. If you have not customized reports before, create a new project by selecting **File | New | Project**.
4. Click **Business Intelligence Projects**.
5. Under Visual Studio Installed Templates, click **Report Server Project**.
6. In **Name**, type MyCustomReports, and then click **OK**.
7. In the Solution Explorer pane, right-click on **Reports** and select **Add | Existing Item**.
8. Locate the 01611a.rdl file in \Microsoft Dynamics\SL\Applications\GL.
9. Double-click to add it, or click to select it and then click **Add**.
10. Open the report.
11. Before making changes, select **Save 01611a.rdl As** on the File menu. Save the file as 01newv1.rdl and store it in \Microsoft Dynamics\SL\Applications\Usr_Rpts.

Note:

- Keep all customized Microsoft Dynamics SL reports in the Usr_Rpts folder.
- Including the 01 prefix in the report name is required for this General Ledger report. Use the appropriate prefix for the module that is associated with the report you are customizing (for example, 08 for an Accounts Receivable report).

12. Modify the report as desired. In this example, two changes are involved.
 - a) The company ID is added to the report header next to the company name using the format name "With Company ID."
 1. Right-click on the `=Code.CmpnyName()` field, and then select **Expression. Edit Expression** opens.
 5. Change the value of the expression to


```
=Code.CmpnyName() + "(" + First(ReportItems!txtCpnyID.Value) + ")"
```
 6. Click **OK**.
 7. Save the report.
 8. On the Edit menu, click **Copy**, and then click **Paste** to create a new copy of the report. The new report appears in Solution Explorer as *Copy of 01newv1.rdl*.
 9. Right-click on Copy of 01newv1.rdl and select **Rename**.
 10. Rename the report *01new2.rdl*.
 - b) Ending balances with negative values are surrounded by parenthesis using the format name "Negative in Prens."
 1. On the **Layout** tab of *01newv1.rdl (Design)*, right-click on the textbox32 field, and then select **Expression**.
 2. Change the expression to


```
=iif ( Fields!Ending_Balance.Value > 0,
Format(Fields!Ending_Balance.Value, "N" &
Code.RIParam("BaseCuryDecPl")), "(" +
Format(Fields!Ending_Balance.Value, "N" &
Code.RIParam("BaseCuryDecPl")) + ")" )
```


3. Click **OK**.
13. Select **Save 01newv1.rdl As** on the File menu, and save this report in Microsoft Dynamics\SL\Applications\Usr_Rpts as 01newv2.rdl. Now there are two formats of the new report, 01newv1 and 01newv2.

Adding a New SSRS Report to the Menu System

Note: When you add a new SSRS report to a menu, be sure to use the program file name of ROISRS.exe.

To add a new SSRS report to the menu system:

1. Create a report control record.
In the System Manager *Report Control Maintenance* (98.300.00) screen, create the control information for the report. Make note of the report number you create so that you can refer to it later. For more information, see “Create a New Report Control Record” on page 55.
2. Add a module record if needed.
If the report is for an existing module, go on to the next step. If the report is for a new module, create a module record in the System Manager *Modules Maintenance* (98.320.00) screen. Make note of the module ID you create so that you can refer to it later. See the System Manager Help or user’s guide for assistance.
3. Create a screen record.
In the System Manager’s *Screen Maintenance* (98.330.00), create a screen record. In **Number**, enter the report number you created in step 1. In **Module**, enter the module ID you created in step 2. In **Type**, select SRS Report. For more information, see “Add the New Report to Existing Screens and Reports” on page 56.
4. Grant users access to the report.
Use the System Manager *Access Rights Maintenance* (95.270.00) screen to grant users access to the report. For more information, see “Add or Modify a Group of Users of the Report” on page 57.
5. Add the report to a menu.
The report can now be seen on the All Modules list in the Microsoft Dynamics SL window. If you want the report to appear on other menus, you must create menus for it. For more information, see “Add the Report to a Menu” on page 59.

User-defined Functions for SSRS Reports

When data is seen on an application screen, it does not appear as it is stored in the database. Since a report draws data directly from the database, it would show the data in its raw form without the availability of user-defined functions. These functions take the raw data as input and return it in its formatted state.

User-Defined Functions have many purposes. For example, you can use them to:

- Format data based on Flexdef definitions.
- Format period numbers as they are in screens.
- Formatting address lines to not leave blank lines when data is incomplete.

How User-Defined Functions Are Created

Two methods are used to create user-defined functions for Microsoft Dynamics SL:

- Using report code.
- Using DynamicsSLFormatting.dll.

How to Use Code-based User-Defined Functions

If you want to use code-based user-defined functions, you will need to write the code yourself, and then add it to the report. You add this code to the report using *Report Properties*.

To create a code-based user-defined function:

1. Go to **Report | Report Properties | Code** tab.

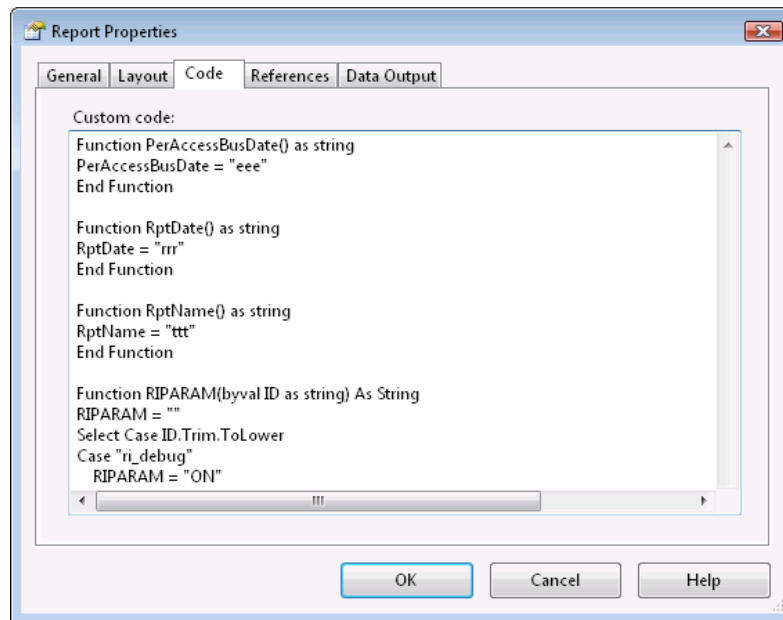


Figure 37: Visual Studio Report Properties, Code tab

2. Define the functions. The code should be written using Visual Basic® syntax.
3. Click **OK** to save the code.

- Once the code is created, add the user-defined functions in the Expression property of the fields created in the report. Right-click on a report field, and then select **Expression**. The *Edit Expression* window appears.

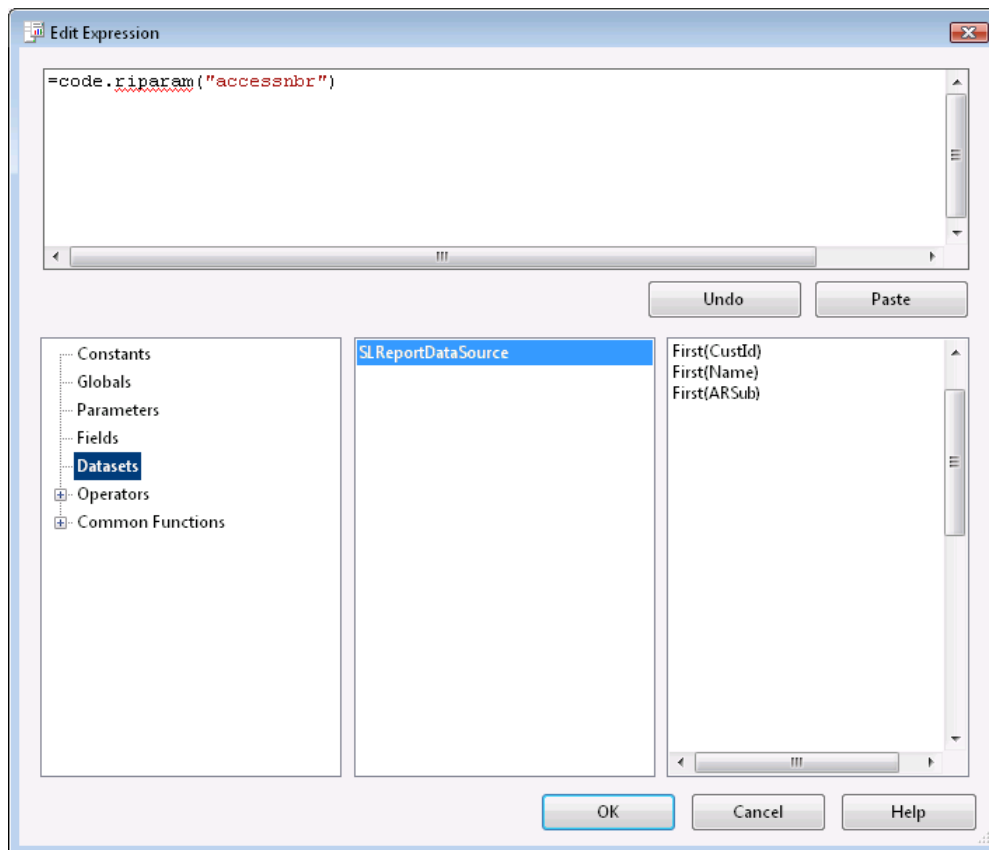


Figure 38: Visual Studio Edit Expression

How to Use Assembly-based User-Defined Functions

If you plan to use a .NET assembly-based user-defined function, you first must add a reference to the assembly into the report.

To create a .NET assembly-based user-defined function:

1. Go to Report | Report Properties | References tab.

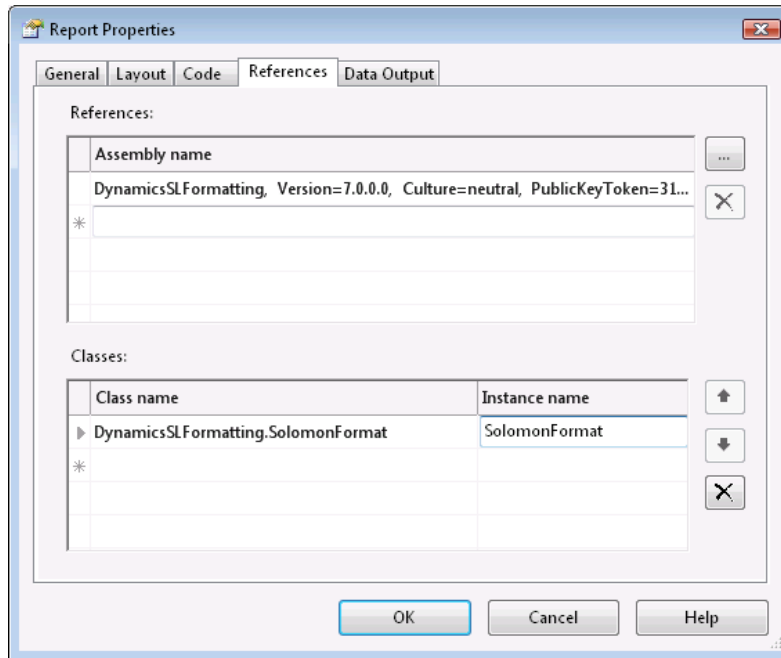



Figure 39: Visual Studio Report Properties, References tab

2. Specify the assembly in **Assembly name**.
 - If you would like to use the user-defined function provided with Microsoft Dynamics SL, click the **Browse** button , and then select DynamicsSLFormatting.dll, which is typically found in \Program Files\Common Files\Microsoft Shared\DynamicsSL. In **Class name**, type DynamicsSLFormatting.SolomonFormat. In **Instance name**, type SolomonFormat.
 - If you plan to use another assembly for user-defined functions, add them using the same basic procedure.

- Use these user-defined functions in the Expression property of the fields in your report. Right-click on a field, and then choose **Expression**. *Edit Expression* appears.

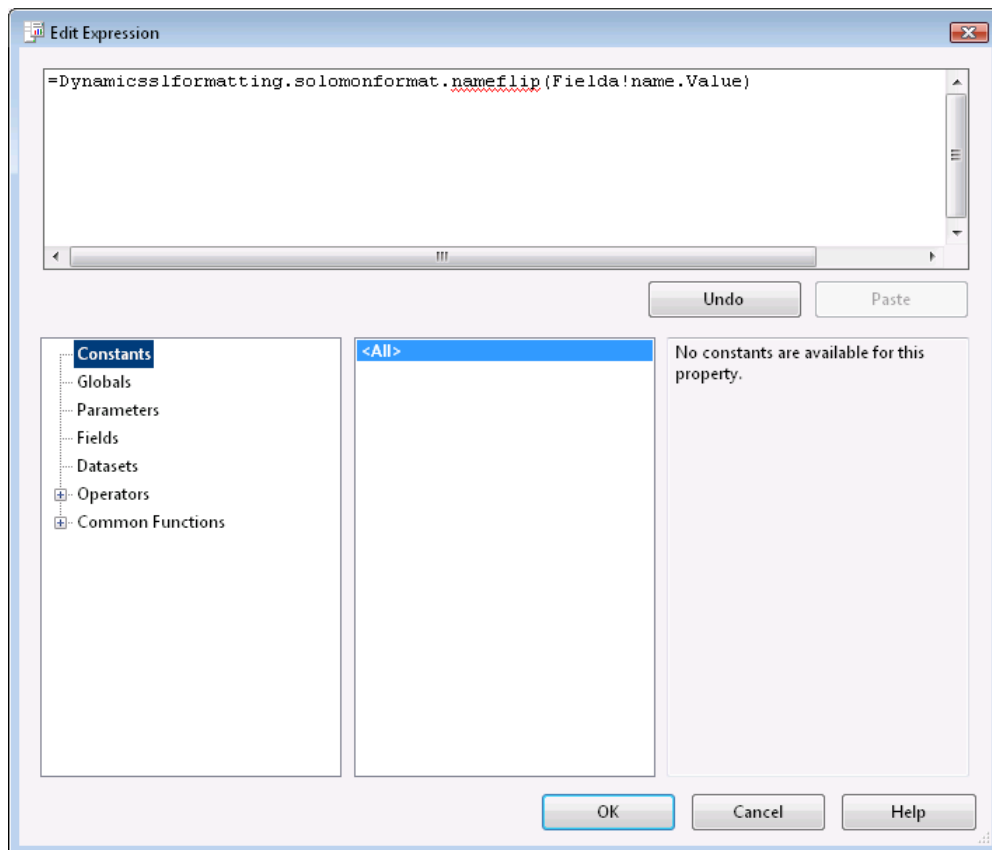


Figure 40: Edit Expression

- To use a user-defined function provided with Microsoft Dynamics SL, enter an expression similar to the one below:

```
=Dynamicsslformatting.solomonformat.nameflip(Fielda!name.Value)
```

Intellisense does not work in the IDE for assembly-based user-defined functions, so you must be familiar with the user-defined functions before you use them.

User-Defined Functions Specific to SSRS Reports

Note: User-defined functions are distributed as .NET assemblies.

DynamicsSLFormatting.dll

The user-defined functions described here deal with address functions. You can use these functions to eliminate the situation in which a printed address has a blank line because it does not contain all elements of a four-line address (for example, there is no second address line). In such a case, AddressLine1 returns the first line of an address, AddressLine2 returns the city, state, and zip code, and the rest of the address follows this pattern.

Example:

```

Addr1 =                1122 Crystal Ave.
Addr2 =
City =                 Fall City
State =                OH
Zip =                  33232
Country/Region =      USA
AddressLine1 will return: 1122 Crystal Ave.
AddressLine2 will return: Fall City, OH 33232
AddressLine3 will return: USA
AddressLine4 will return:

```

AddressLine1(String, String, String, String, String, String)

Description

Obtains the first line of an address that consists of Addr1, Addr2, City, State, Zip, Country/Region, where any line can be empty.

Syntax

Any specific requirements for the input parameters—like the fact that FmtDate() only accepts date formats of mm/dd/yy or mm/dd/yyyy, or limits on the lengths of input strings—should be listed.

AddressLine2(String, String, String, String, String, String)

Description

Obtains the second line of an address that consists of Addr1, Addr2, City, State, Zip, Country/Region, where any line can be empty.

Syntax

```
AddressLine2(addr1, addr2, city, state, zip, country/region)
```

AddressLine3(String, String, String, String, String, String)

Description

Obtains the third line of an address that consists of Addr1, Addr2, City, State, Zip, Country/Region, where any line can be empty.

Syntax

```
AddressLine3(addr1, addr2, city, state, zip, country/region)
```

AddressLine4(String, String, String, String, String, String)*Description*

Obtains the fourth line of an address that consists of Addr1, Addr2, City, State, Zip, Country/Region, where any line can be empty.

Syntax

```
AddressLine4(addr1, addr2, city, state, zip, country/region)
```

FmtDate(String)*Description*

Accepts a date string of format *mm/dd/yy* or *mm/dd/yyyy* and converts the string to the Crystal Reports internal date format.

Syntax

```
FmtDate(date)
```

FmtPeriod(String)*Description*

Returns a period number in the format *pp-yyyy*.

Syntax

```
FmtPeriod(PeriodValue)
```

Example

```
FmtPeriod(Fields!ArTRan.CurPer)
```

FmtPhone(String)*Description*

Returns the telephone number in the format (999)999-9999 Ext 99999.

Syntax

```
FmtPhone({Vendor.phone})
```

FmtScrnNbr(String)*Description*

Returns the screen number (Microsoft Dynamics SL window number) in the format 99.999.

Syntax

```
FmtScrnNbr(Screen)
```

Example

```
FmtScrnNbr("99999")
```

FmtZip(String)*Description*

Returns the zip code in the format 99999-9999.

Syntax

```
FmtZip({Address.Zip})
```


NameFlip(String)

Description

Returns a string similar to the source string except the contents are flipped around an embedded ~ character.

Syntax

```
NameFlip(name)
```

Example

```
NameFlip ("Bank of Baltimore~The")
```

This returns The Bank of Baltimore.

Transform(SourceString, FormatString)

Description

Transforms the source string into a string that matches the format specified in the format string.

Syntax

```
Transform(expr, format)
```

Example

```
Transform("0000000000", "@RXX-XXX-XXX-XX")
```

This returns 00-000-000-00.

Special characters in the mask include:

- X – Letter
- x – Letter
- ! – Capitalize the letter

For all other characters, the value of the mask becomes part of the result.

TransformNbr(Number, String)

Description

Transforms the number into a string that matches the format specified in the format string.

Syntax

```
TransformNbr(Number, String)
```

Example

```
TransformNbr(45.99, "$99.9")
```

This returns \$46.0.

Special characters in the mask include:

- \$ – Appears as is
- * – Appears as is
- Others ignored
- 9 – Number

TransformNbrLocale ()

Description

Transforms the number into a string using the localization options identified in FormatNumber.

Syntax

```
TransformNbrLocale (Number, FormatNumber)
```

Possible FormatNumber values are:

- 0 – Use current system locale
- 1 – Use locale decimal precision and decimal separator value
- 2 – Use locale thousands grouping and thousands separator value
- 4 – Format number as a negative value using system locale settings
- 16 – Use the currency settings for the current system locale

Possible values 1, 2, and 4 can be combined as in the following examples.

Example

```
TransformNumberLocal (12345.678, 0) returns 12,345.68 as the number format.  
TransformNumberLocal (12345.678, 1) returns 12345.68 as the number format.  
TransformNumberLocal (12345.678, 2) returns 12,346 as the number format.  
TransformNumberLocal (12345.678, 1+2) returns 12,345.68 as the number format.  
TransformNumberLocal (12345.678, 1+2+4) returns (12,345.68) as the number  
format.  
TransformNumberLocal (12345.678, 16) returns $12,345.68 as the number format.
```

Report Code

SL_Query

Description

Returns the SQL statement that the report is using to generate the report.

Syntax

```
Code.SL_Query
```

CpnyName

Description

Returns the company name of the current company in which the user is logged on.

Syntax

```
Code.CpnyName
```

FRW1Per

Description

Returns a string in the format *For the Period XX Through YY* (XX is RptRuntime, Longanswer00; YY is RptRuntime, Longanswer01).

Syntax

```
Code.FRW1Per ()
```

FRW2Per*Description*

Returns a string in the format *Period XX As of YY* (XX is RptRuntime, Longanswer02; YY is RptRuntime, Longanswer01).

Syntax

```
Code.FRW2Per ( )
```

FRW3Per*Description*

Returns a string in the format *Fiscal Year XX* (XX is RptRuntime.begPernbr [last four digits]).

Syntax

```
Code.FRW3Per ( )
```

RIPARAM(String)*Description*

Retrieves a value from the current RIPARAM memory area.

Syntax

```
Code.RIPARAM(string)
```

Possible string value is:

RptNbr – Returns the Report Number

RptDate*Description*

Returns the report date in the format *Periods: (Beginning Period Number) Through (Ending Period Number) Period: (Beginning Period Number) As of: (Report Date)* with the beginning and ending period numbers in the format 99-99.

Syntax

```
Code.RptDate
```

RptName*Description*

Returns the name of the currently running report (this name is referred to off the menu).

Syntax

```
Code.RptName
```

ReptNumber*Description*

Returns the report number in the format *Rept 99.999*.

Syntax

```
Code.ReptNumber
```

PerAccessBusDate

Description

Returns the period number, the access number, and the business date (if the system date equals the business date). Uses the format *(Period Number)-(Access Number) Bus: (Business Date)* with the period number in the format 99-99.

Syntax

```
Code.PerAccessBusDate
```

ROISRS RIParams

Report information parameters (RIParams) are report items defined by the ROISRS that control the type and length of the data printed on reports (for example, the report title and the number of characters in the report title).

ROISRS reporting for Microsoft Dynamics SL uses two types of RIParam: standard Microsoft Dynamics SL RIParams and additional (custom) RIParams.

Note: All Microsoft Dynamics SL RIParams are display-only report items.

RIParams are implemented using the Report Code feature of a SSRS report. At runtime, the ROISRS will rewrite the RIPARAM() function in the code of the report that is being processed.

Microsoft Dynamics SL RIParams

The following table lists the RIParams used for standard Microsoft Dynamics SL reporting. The ROISRS automatically sets the value for each of these RIParams.

Name	Data Type * Length
AccessNbr	As String * 5
Acct	As String * 10
Banner	As String * 1
BatNbr	As String * 10
BegPerNbr	As String * 6
BusinessDate	As String * 10
CmpnyName	As String * 30
ComputerName	As String * 21
CpnyID	Char * 10
DatabaseName	As String * 20
DocNbr	As String * 10
EndPerNbr	As String * 6
LongAnswer(0 To 4)	As String * 80
NotesOn	As Integer
PerNbr	As String * 6
RepBegDate	As String * 10
RepEndDate	As String * 10
ReportDate	As String * 10
ReportFormat	As String * 30
ReportName	As String * 30
ReportNbr	As String * 5
ReportTitle	As String * 30
RI_BEGPAGE	As Integer
RI_CHKTIME	As String * 1
RI_COPIES	As Integer
RI_DATADIR	As String * 21
RI_DICTDIR	As String * 21
RI_DISABLEDSDS	As SmallInt
RI_DISABLEQS	As SmallInt
RI_DISPERR	As String * 1
RI_ENDPAGE	As Integer

Name	Data Type * Length
RI_ID	As Integer
RI_INCLUDE	As String * 1
RI_LIBRARY	As String * 80
RI_NOESC	As String * 1
RI_OUTAPPN	As String * 1
RI_OUTFILE	As String * 80
RI_PRINTER	As String * 1
RI_REPLACE	As String * 255
RI_REPORT	As String * 30
RI_STATUS	As String * 1
RI_TEST	As String * 1
RI_WHERE	As String * 255
RI_WPORT	As String * 40
RI_WPTR	As String * 40
RI_WTITLE	As String * 30
SegCustMask	As String * 16
SegCustTitle	As String * 16
SegInvenMask	As String * 26
SegInvenTitle	As String * 26
SegSubMask	As String * 38
SegSubTitle	As String * 38
SegVendMask	As String * 16
SegVendTitle	As String * 16
ShortAnswer(0 To 4)	As String * 10
Sub	As String * 24
SystemDate	As String * 10
SystemTime	As String * 7
UserId	As String * 47

ROISRS Standard RIParams

Standard RIParams are pre-defined parameters that you can insert in a pre-process to control certain aspects of report printing. For example, you might want to insert **RI_FONTITALIC** to specify that a report should be printed using the *italic* font style.

RIParams could be individual fields from setup records or any other database records, or they could be bits of information you want the report to access.

Available RIParams are listed below.

RI_DEBUG

Determines if Microsoft.Dynamics.SL.Reporting.dll is in Debug mode.

RI_DRIVER

Identifies the name of the printer driver minus the .drv extension.

RI_FILETYPE

Identifies the type of file the ROISRS creates when exporting a report.

Value	Description
0	Acrobat
8	Excel

The ROISRS accesses this information and modifies the report at runtime.

RI_FONTBOLD

Indicates whether or not the font identified by **RI_FONTNAME** should also be set in **bold** font style.

Value	Description
0	Do not set the font to boldface
1	Set the font to boldface

Microsoft.Dynamics.SL.Reporting.dll accesses this information and modifies the report at runtime.

RI_FONTITALIC

Indicates whether or not the font identified by **RI_FONTNAME** should also be set in *italic* font style.

Value	Description
0	Do not set the font to italic
1	Set the font to italic

Microsoft.Dynamics.SL.Reporting.dll accesses this information and modifies the report at runtime.

RI_FONTNAME

Identifies the font name used by the report, either an actual font name or the text {Report Font}.

Microsoft.Dynamics.SL.Reporting.dll accesses this information and modifies the report at runtime. If {Report Font} appears in the RIParam, no font modifications are performed on the current report.

RI_FONTSIZE

Identifies the font size used by the report. Microsoft Dynamics.SL.Reporting.dll accesses this information and modifies the report at runtime, dividing the font size value by 10 and using the result for report printing. This is acceptable because SWIM takes the value from the *Font* window (accessed by clicking **Fonts** in *Printer Options* (98.220.00)), and multiplies this value by 10 before storing the result.

RI_ORIENT

Indicates the printing orientation of the printer identified in *Printer Options* (98.220.00).

Value	Description
1	Print pages in the portrait orientation
2	Print pages in the landscape orientation

Microsoft.Dynamics.SL.Reporting.dll accesses this information and modifies the report at runtime.

RI_USEPORIENT

Tells Microsoft.Dynamics.SL.Reporting.dll whether to print the report based on the report's orientation or the printer's orientation.

Value	Description
0	Use the report's orientation
-1	Use the printer's orientation

New RIParams

For information about this topic, see "Creating New RIParams" on page 72.

Screens

Printer Options (98.220.00)

Use to select a printer other than the default, to send a report to a file or Microsoft® SharePoint® document library, or to establish the default printing options. Default printing options include the print destination, the font in which all information is generated, and the orientation of the report on the printed page.

Note: You can select default settings for the options on this screen using the [Print Default] section of the Solomon.ini file. For more information about the Solomon.ini file, see the System Manager Help or user's guide.

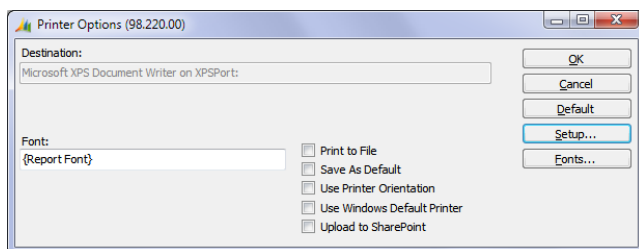


Figure 41: Printer Options (98.220.00)

Following are the field descriptions for *Printer Options (98.220.00)*.

Destination

Specifies where the software sends information when printing.

- If the software is sending information to a printer, this field displays the name of the printer and the port to which it is attached.
- If the software is sending information to a file, this field displays the path and file name to the location where the information is being sent.
- If the software is sending information to a SharePoint document library, this field displays the path of the user's TEMP environment variable.

Font

The default printer font, which includes typeface, style, and size that the system uses when generating reports.

Print to File (check box)

Select to send report information to a file rather than to a printer. The options are:

- Acrobat (PDF)*
- Crystal Reports
- HTML 3.2
- HTML 4.0, DHTML
- Excel 97-2003*
- Excel 97-2003 (data only)
- Microsoft Office Word
- Rich Text
- Text
- XML
- Tab-separated text

* Supported by SQL Server Reporting Services.

Save as Default (check box)

Saves the selected printing options as the default printing options.

Use Printer Orientation (check box)

Specifies whether the report will be printed using the page orientation of the current printer. For example, if the check box is selected and the current printer is set for portrait orientation, the report prints in portrait orientation, even if the report layout indicates landscape orientation. If the check box is not selected, the report prints using its own page orientation rather than that of the printer.

Use Windows Default Printer (check box)

Tells the software to use the default printer defined for the Windows operating system instead of the default printer.

Upload to SharePoint (check box)

Sends information to a Microsoft® SharePoint® document library. Customers, employees, vendors, and projects setup for Quick Send or Doc Share are excluded from the file uploaded to SharePoint if Quick Send and Doc Share were not disabled at the time the report was generated.

Default (button)

Click **Default** to set the printing options to the default values.

Setup (button)

Click **Setup** to open the *Print Setup* dialog box and configure the printer or specify the full path to the location where Microsoft Dynamics SL saves the report information.

Fonts (button)

Click **Fonts** to open the *Fonts* dialog box and define the type specifications of the printer font.

Upload to SharePoint Document Library (98.220.01)

Use to define the Microsoft® SharePoint® document library location when reports are published to SharePoint sites.

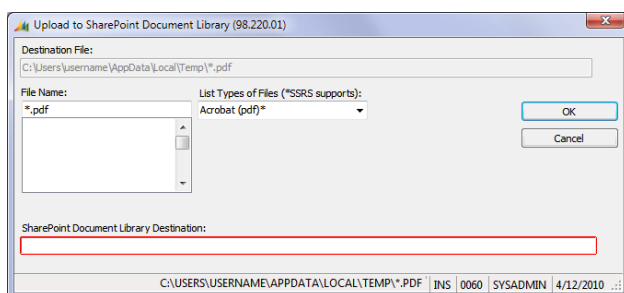


Figure 42: Upload to SharePoint Document Library

For more information, see the System Manager Help or user's guide.

Following are the field descriptions for *Upload to SharePoint Document Library (98.220.01)*.

Destination File

The full path of the user's TEMP environment variable. A report is sent to a file in this directory prior to being published to a SharePoint document library.

File Name

The name of the file that will hold the report information; if left as *.pdf, the file name is automatically assigned based on the report screen number.

List Types of Files

The type of file that will hold the report information; the .pdf file type is the default.

SharePoint Document Library Destination

The full path to the SharePoint document library. Press F3 to open the *Document Library Search* dialogue box which displays previously accessed SharePoint document libraries.

Document Library Search

Use to display document libraries on SharePoint sites.

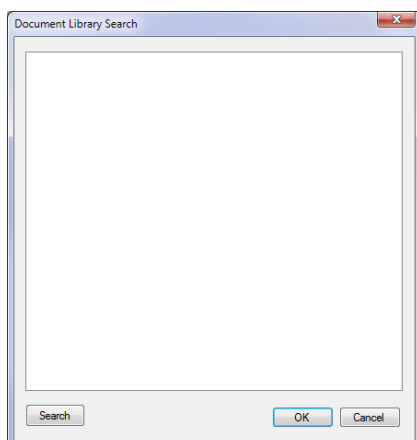


Figure 43: Document Library Search

Following is the field description for *Document Library Search*.

Search (button)

Opens the *Document Libraries* dialogue box which allows the entry of a SharePoint site path so that its document libraries are displayed on *Document Library Search*.

Document Libraries

Use to specify the path to a SharePoint site so that its document libraries are retrieved and displayed on the *Document Library Search* dialogue box.

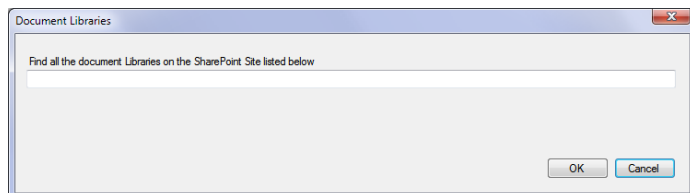


Figure 44: Document Libraries

Following is the field description for *Document Libraries*.

Find all the document Libraries on the SharePoint Site listed below

The full path to a Microsoft® SharePoint® site so that its document libraries are displayed on the *Document Library Search* dialogue box (for example, <http://servername>).

ROI Screen

Options on the Report Option Interpreter (ROI) screen vary based on the module the report is generated in, as well as the report itself.

ROI Screen, Report Tab

Use the **Report** tab to define options that control the appearance and content of the standard report. For example you might print a report in summary or detail on plain paper or a pre-printed form. For more information, see “Report Tab Settings” on page 6.

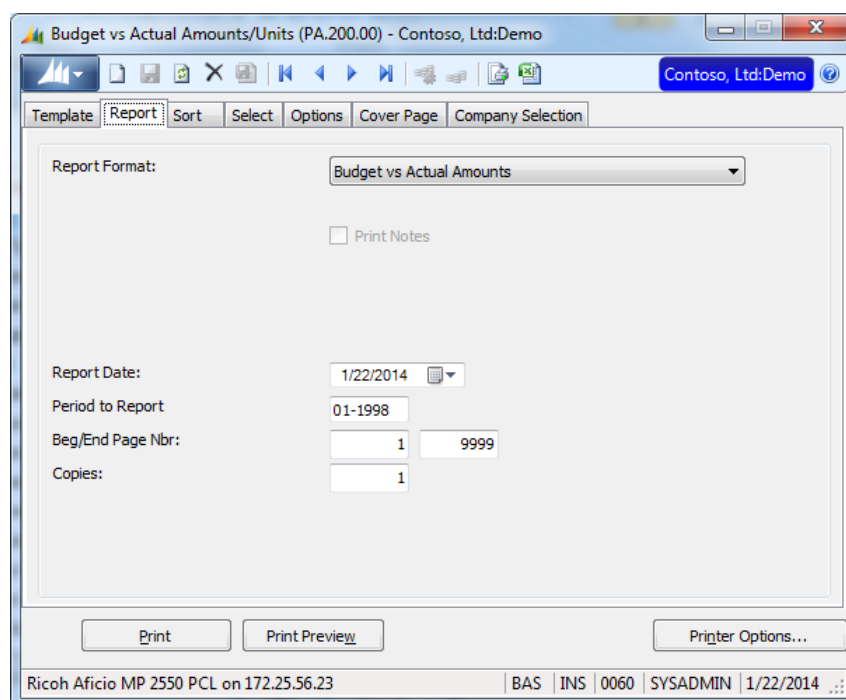


Figure 45: Report tab

Following are field descriptions for the **Report** tab.

Report Format

Controls overall report output for such areas as level of information (summary or detail records), amount of information (standard amounts only or standard and multi-currency amounts), and type of output (plain paper or preprinted forms). Many reports have multiple report format options.

Print Notes

Includes any notes (attached to data records in windows) on the report. Not every report has the capability to print notes.

Do not send electronically (check box)

Select to include all data items in the report. Quick Send preferences defined for customers, employees, projects, and vendors are ignored. This check box appears when document types are defined on *Quick Send Setup* (21.951.00) in the Shared Information module. For more information about setting up Quick Send, see “Setting up Quick Send” in the Shared Information online Help or user’s guide.

Do not publish Doc Share requests to SharePoint (check box)

Select to include all data items in the report. Doc Share settings defined for customers, projects, and vendors are ignored. This check box appears when entity types are configured on *SharePoint Site Configuration* (98.360.00). For more information about setting up Doc Share, see “Sharing Documents Using Doc Share” in the System Manager online Help or user’s guide.

Report Date

Identifies the date when the report was printed. This date appears in the report header and is typically the current date.

Beg/End Date

Identifies the first and last date in a range of dates for which information is included on the report. **Beg/End Date** only appears for certain reports.

Beg/End Period

Identifies the first and last periods in a range of fiscal periods whose information should be included on the report. **Beg/End Period** only appears on certain reports.

Period to Report

Identifies a specific fiscal period whose information should be included on the report. **Period to Report** only appears on certain reports.

Fiscal Year

Identifies a specific fiscal year whose information should be included on the report. **Fiscal Year** only appears on certain reports.

Beg/End Page Nbr

Identifies the first and last pages in a range of report pages whose information should be included on the report.

Note: When you preview a report on screen (click **Print Preview**), the preview shows all pages in the report even if you have selected a specific page range to print. When you send the report to the printer (click **Print**), only those pages you have identified are actually printed.

Copies

Controls how many copies of the report are actually printed after you start the printing process.

ROI Screen, Template Tab

Use the **Template** tab to define a template of the report's current print settings (for example, the options selected in all other ROI tabs) in order to speed up printing the report at a later time. Defining a report template saves the current print settings in the database under a unique template ID and description. For more information, see "Template Tab Settings" on page 8.

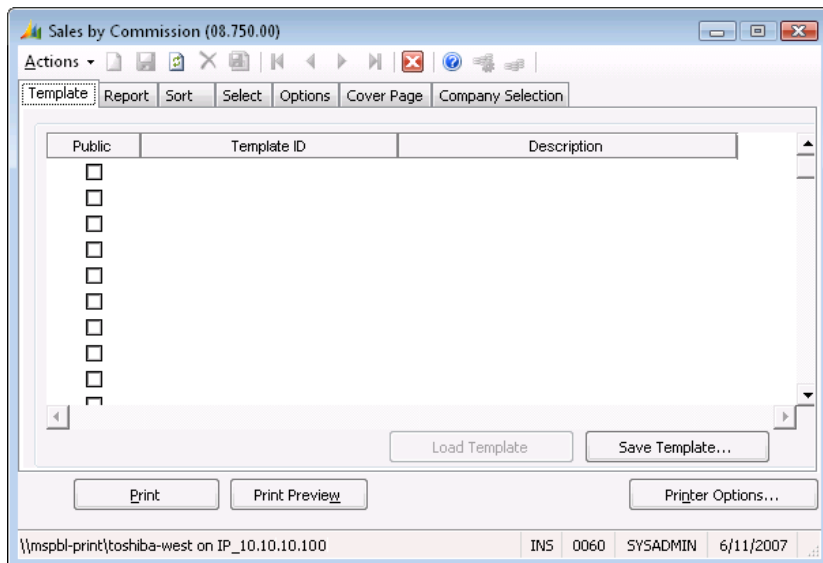


Figure 46: Template tab

Following are field descriptions for the **Template** tab.

Public

Controls whether all Microsoft Dynamics SL users can use the template or only the user who defined the template (this is controlled by user ID). Select **Public** to give all Microsoft Dynamics SL users access to the template.

If a template is private (**Public** not selected), it will not appear on the **Template** tab list when users other than the one who defined the template are printing the template's report.

Template ID

The ID of the report template. Use a template's ID to print the report according to the template's print settings.

Description

A description of the template's reporting purpose (for example, print the report in summary for periods 03 through 06).

Load Template (button)

Tells the software to use the print settings of the template that is currently selected when printing the report.

Save Template (button)

Saves the current report print settings to the application database under the report ID and description defined at **Template ID** and **Description**.

Note: You can also save the report template information by choosing **Actions | Template** from the menu.

ROI Screen, Sort Tab

Use the **Sort** tab to define a custom sort order for report information based on any of the report's record.filename fields. For example, the standard sort order priority for customer names is last name first, first name second. With the **Sort** tab, you could set up a report to sort customers based on the sort order first name first. Or you might generate a report that sorts customers based on year-to-date net sales. For more information, see "Sort Tab Settings" on page 10.

Note: The available custom sort options depend largely on the structure of the report. Because of this, a report created using custom sort options may not appear as you intend. To ensure that a custom-sorted report will print as you expect, preview the report using the custom sort options before sending the report to the printer.

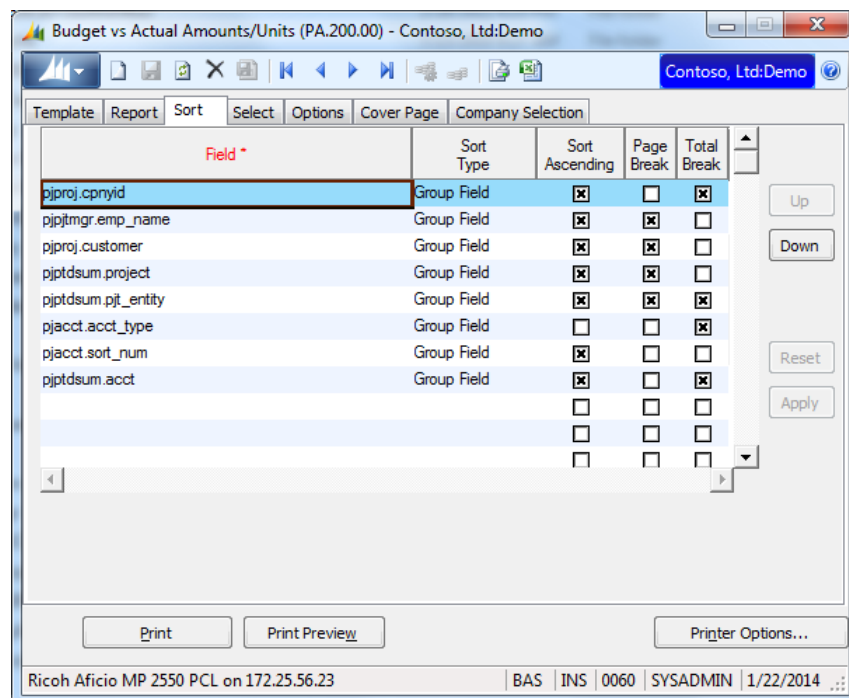


Figure 47: Sort tab

Following are field descriptions for the **Sort** tab.

Field

Identifies the database record.filename on which to base the report's custom sort order. Press **F3** to open a *Possible Values* (PV) window listing all record.fieldnames for the report. Select a record.filename, and then click **OK**.

Note: **Field** is a combination of table name and field name. For example, *gltran.refnbr* indicates that the field refnbr is found in the GLTran table. To learn more about the fields on the **Sort** tab, open the online help, and select **Help Topics | Schema | Full Schema**. Choose the schema topic for the module's series (for example, Financial Management Schema). Select the table name, and then search for the specific field name.

Sort Type

Describes the type of field identified in **Field**. The options are Group Field or Sort Field. Group Field indicates that the field is a report group that allows page and total breaks. Sort Field indicates that the report data is to be sorted by the values in the field within existing groups. Group fields take precedence over sort fields and are processed first.

SQL Server Reporting Services (SSRS) allows a report developer to add more than one set or type of grouping in a report using tables, lists, matrices, or rectangles. The ROISRS supports the modification of groups only when one type of grouping is used in a report. Also, the ROISRS only supports the modification of table and list groupings. If the ROISRS detects more than one type of grouping or multiple group sets in a report, no grouping information will appear and modifications or additions of group field type detail lines will not be allowed.

Sort Ascending

Determines how the data of the record.fieldname selected at **Field** should be sorted on the report, in ascending order (1, 2, 3, etc.) or descending order (99, 98, 97, etc.). Select **Sort Ascending** to sort in ascending order. Clear **Sort Ascending** to sort in descending order.

Page Break

Determines if the software should insert a page break every time the value specified by record.fieldname at **Field** changes (for example, if the sort order is based on customer name, insert a page break between each new customer name on the report). Select **Page Break** to insert a page break between each new record.fieldname value. Clear **Page Break** to omit the page break.

Total Break

Determines if the software should insert a total break every time the value specified by record.fieldname at **Field** changes. Select **Total Break** to insert a total break between each new record.fieldname value. Clear **Total Break** to omit the total break.

Note: This is not supported in SSRS reports. Changes to this field using the ROISRS will be ignored.

Up (button)

The **Up** button is used to move the selected row up.

Down (button)

The **Down** button is used to move the selected row down.

Reset (button)

The **Reset** button is used to restore the grouping and sorting criteria from the report. This allows you to reset options to their original values after you have changed them.

Apply (button)

The **Apply** button is used to apply changes for this specific report generation. It also refreshes the display to show values in the order they will be used.

ROI Screen, Select Tab

Use the **Select** tab to print a report that contains only a subset of the report's available information (for example, only a portion of the total information maintained for that report in the database). For example, instead of printing all vendor information, you might want to print a *Vendors* (03.670.00) report listing only those vendors located in Ohio. Or you might want to print a *Customer* (08.650.00) report listing only customers with year-to-date sales of \$50,000 or more. For more information, go to "Select Tab Settings" on page 12.

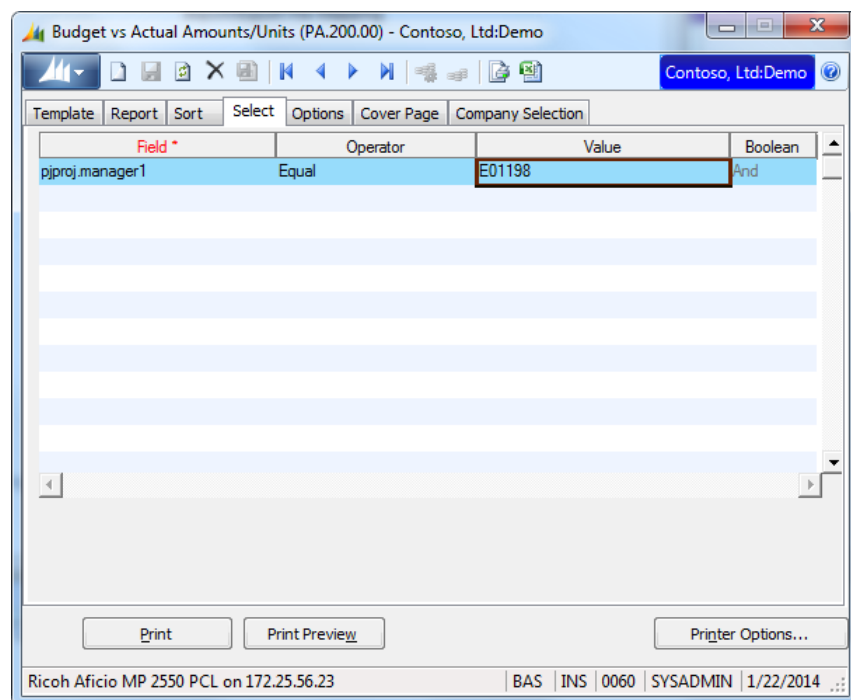


Figure 48: Select tab

Following are field descriptions for the **Select** tab.

Field

Identifies the database record.fieldname to use for selecting report information to be printed. Press F3 to open a *Possible Values* (PV) window listing all record.fieldnames for the report. Select a record.fieldname, and then click **OK**.

Operator

The expression that is used in the selection. For more information see "Value" below. The options are as follows:

- **Between** – Type in values separated by comma or press F3 and the *Between* screen appears for you to enter values. You can press F3 in the *Between* screen fields to select from values in the database or a date on a calendar.
- **Contains** – **Value** only accepts ad-hoc values.
- **Equal** – Type in **Value** or press F3. The calendar appears if the field is a date, otherwise it displays values in the database.
- **Greater than** – Type in **Value** or press F3. The calendar appears if the field is a date, otherwise it displays values in the database.
- **Greater than or equal to** - Type in **Value** or press F3. The calendar will appear if the field is a date, otherwise it will display values in the database.

- In - List of values. Type in values separated by a comma(,), or press F3 and select from a value in the database.
- Is NULL -No **Value** required.
- Less than - Type in **Value** or press F3. The calendar will appear if the field is a date, otherwise it will display values in the database.
- Less than or equal to - Type in **Value** or press F3. The calendar will appear if the field is a date, otherwise it will display values in the database.
- Not between -Type in values separated by comma or press F3 and *Between* screen appears for values to be entered. You can press F3 in the *Between* screen fields to select from values in the database or a date on a calendar.
- Not contains - **Value** will only accept ad-hoc values.
- Not equal- Type in **Value** or press F3. The calendar will appear if the field is a date, otherwise it will display values in the database.
- Not in - Not in the list of values, type in values separated by a comma(,), or press F3 and select from a value in the database.
- Is not NULL - No **Value** required.

Value

The record.fieldname value (for example, amount, ID, account number, etc.) on which you are basing the report selection. For example, if you select Account.ACCT at **Field** and Greater than at **Operator**, typing 1010 here would print a report containing database records with account numbers greater than 1010. See “Operator” above for explanations of how you can populate **Value**.

Press F3 to open screens which help with the selection of dates or values.

Note: If you use an operator like *Between*, that requires more than one report select value, type a comma or the word *and* between the values that you enter (*1010,2020* or *1010 and 2020* for example).

Between and Not Between

Between will appear when you press F3 in the **Value** field after the operator *Between* or *Not Between*. There is a text box for the beginning and ending values or dates to be entered. You can press F3 to select a date or value in the database in these boxes.

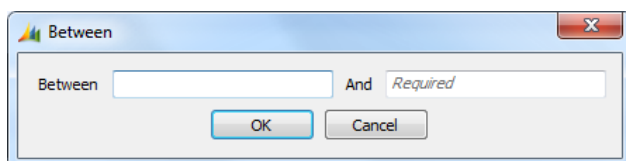


Figure 49: *Between* used with values in the database

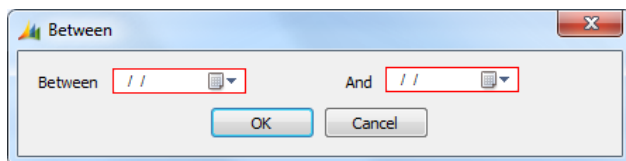


Figure 50: *Between* used with a date field

Equal, Greater Than, Greater than or equal to, Less than, Less than or equal to, and Not Equal

Value List appears for most data types when you press F3 and Equal, Greater Than, Greater than or equal to, Less than, Less than or equal to, and Not Equal are used as the operator. Click on the value that you want selected. You can filter the list by entering partial values in the filter box.

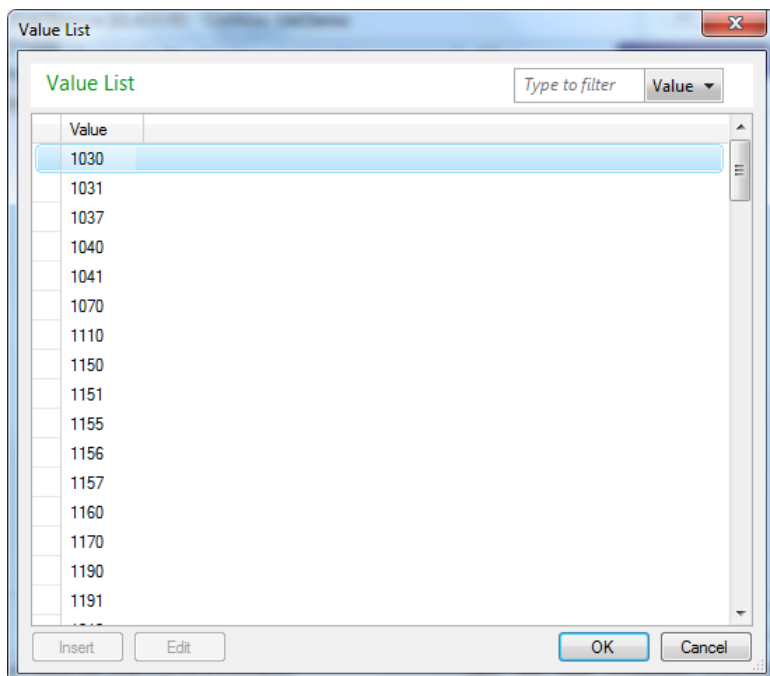


Figure 51: Value List

In and Not in

In will appear when you press F3 in **Value** for most data types when you select *In* or *Not in* as the operator. Click **Add to List** to select those items you want included. Type a value in **Enter a Value** if it is not in the list that appears, and then click **Add to List**. Click **Remove from List** to remove an item. To remove everything from the list, click **Clear List**. You can filter the list by entering partial values in the filter box. When the list includes all the values that you want to include, click **OK**.

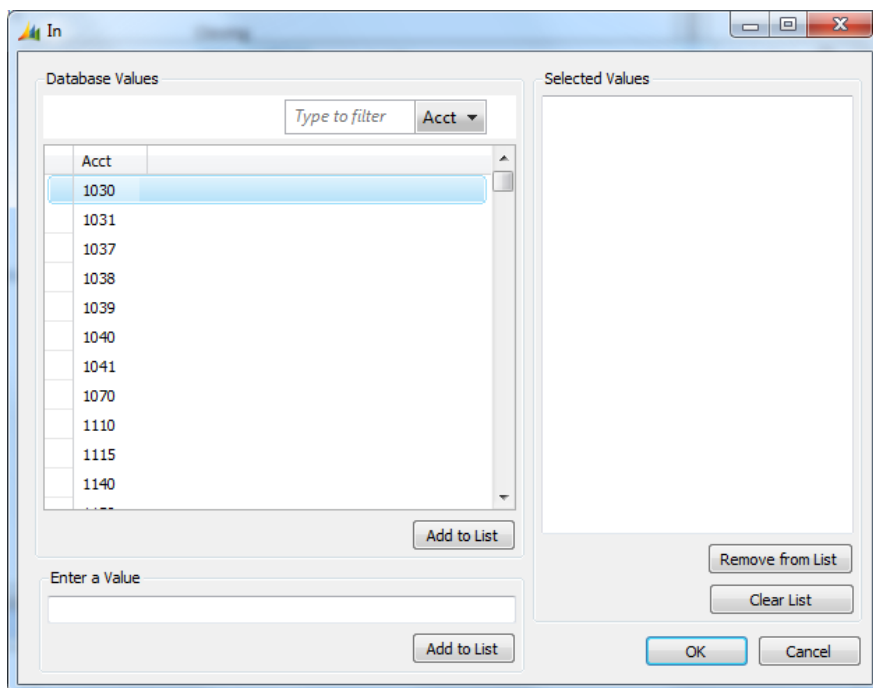


Figure 52: In

Boolean

Boolean operators specifying the logical condition between two or more report selection criteria (for example, the conditions specified by **Field**, **Operator**, and **Value**). The **Boolean** options are:

- **AND** All search conditions connected by AND must be true.
- **OR** At least one of the search conditions connected by OR must be true.

Note: When the **Select** tab is in grid view, an AND default displays under Boolean in the first selection criteria line. This does not represent the logical condition between the first and second lines of user-defined report selection criteria. Instead, it is the logical condition between the report's default selection criteria and the first line of user-defined selection criteria. To AND the first and second lines of selection criteria, select AND at the end of the second line (for example, post-fix the notation).

ROI Screen, Options Tab

Use the **Options** tab to select report-specific additional report options that further control the content of reports. For example, you might use options on the **Options** tab to exclude tax information, age documents by invoice date, or include trade discount amounts. For more information, go to “Options Tab Settings” on page 13.

Note: If you are a developer, you can use the **Options** tab to modify the additional report options of the standard reports or add as many as five additional report options to the custom reports you develop. You can also use the **Options** tab to make a report's information available to other processes. For more information, see “**Advanced Reporting Options**”

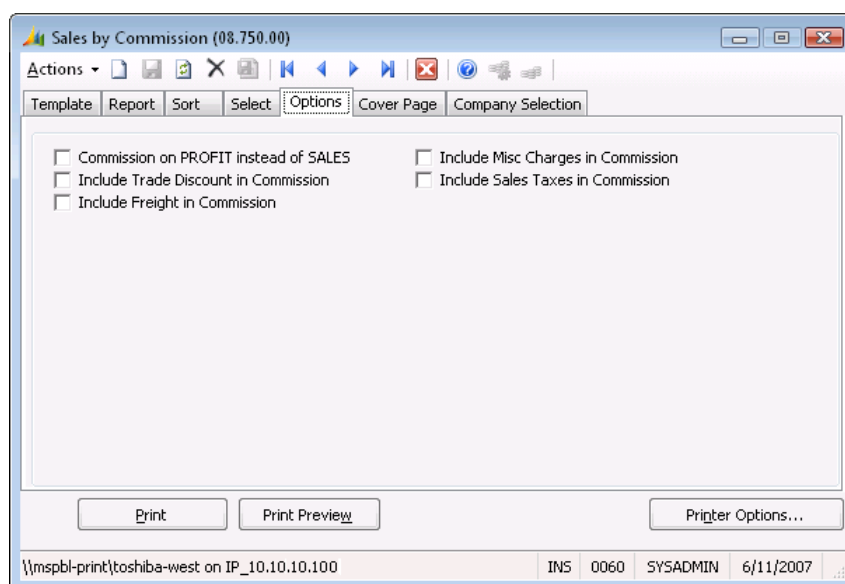


Figure 53: Options tab

The fields on the **Options** tab vary, depending on the purpose of the report. For an explanation of each field on the **Options** tab, see the appropriate Help or user's guide.

ROI Screen, Cover Page Tab

Use the **Cover Page** tab to print a cover page for the current report. A report cover page is an introduction to the report, providing a summary of the report's content, selection criteria, and sort order. It also contains any user-defined comments defined in the **Cover Page** tab prior to printing the report. For more information, see "Cover Page Tab Settings" on page 14.

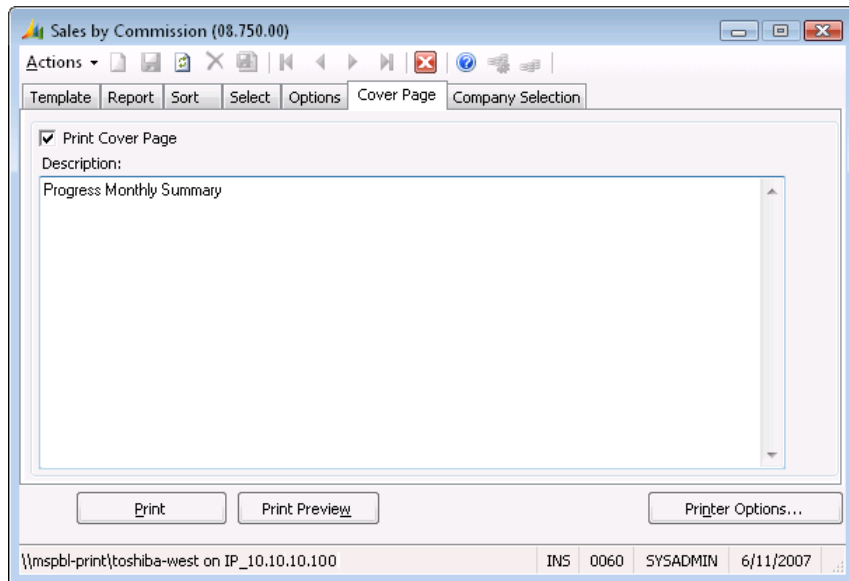


Figure 54: Cover Page tab

Following are field descriptions for the **Cover Page** tab.

Print Cover Page

Determines whether or not ROI prints a cover page report before printing the current report. Select **Print Cover Page** to print a cover page report.

Description

User-defined text to be printed on the cover page report along with report content, sort, and select information.

ROI Screen, Company Selection Tab

The **Company Selection** tab is available if **Multi-Company Selection Allowed** is selected on *Control Options* (98.300.03), which can be accessed from the *System Manager Report Control Maintenance* (98.300.00) screen. The **Company Selection** tab includes options to help you print reports in the following formats:

- A single report for the company in which you are currently working
- A combined report for companies that share the same application database
- Individual reports for companies that share the same system database

For more information, see “Company Selection Tab Settings” on page 15.

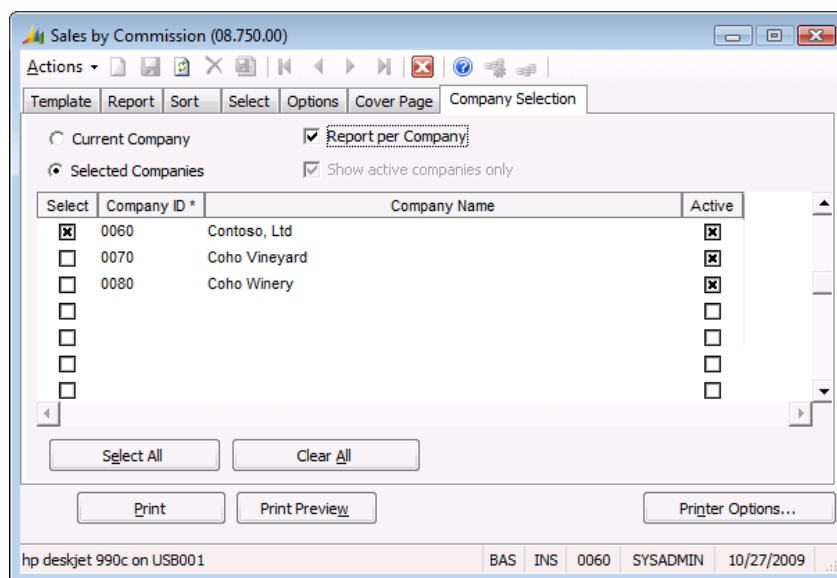


Figure 55: Company Selection tab

Following are field descriptions for the **Company Selection** tab.

Current Company

Select this option to print a report for the company whose database is currently open. This is the default selection.

Selected Companies

Select this option to print a combined report for all companies you choose from the list in the grid area. Only companies that share the same application database appear.

Report per Company (check box)

Using this option, you can print an individual report for each company you choose. Selecting this option also automatically activates **Selected Companies** so you can view and select from a list of companies in the grid area. Only companies that share the same system database appear.

Show active companies only (check box)

Determines whether or not the **Company Selection** tab grid lists currently active companies only. Select **Show active companies only** to list only active companies.

Select

Determines whether or not a company is included in the multi-company report. Choose **Select** to include the company in the report.

Company ID

The ID of a company that can be included in the report.

Company Name

The name of a company that can be included in the report.

Active

Determines whether or not company will be identified as active on the report. Select **Active** to identify the company as active.

Select All (button)

Selects all available companies for inclusion on the report.

Clear All (button)

Resets the **Company Selection** tab grid so that no companies are selected.

Field List

Field List will display after pressing F3 to open a *Possible Values (PV)* window, listing all record.fieldnames for the report. You can sort based on the field selected in the drop-down box. The sorting can also limit what appears in the screen by entering a partial list in the box.

If you want to display only those fields on the report, select Report Field in the drop-down box and type Y in the text box.

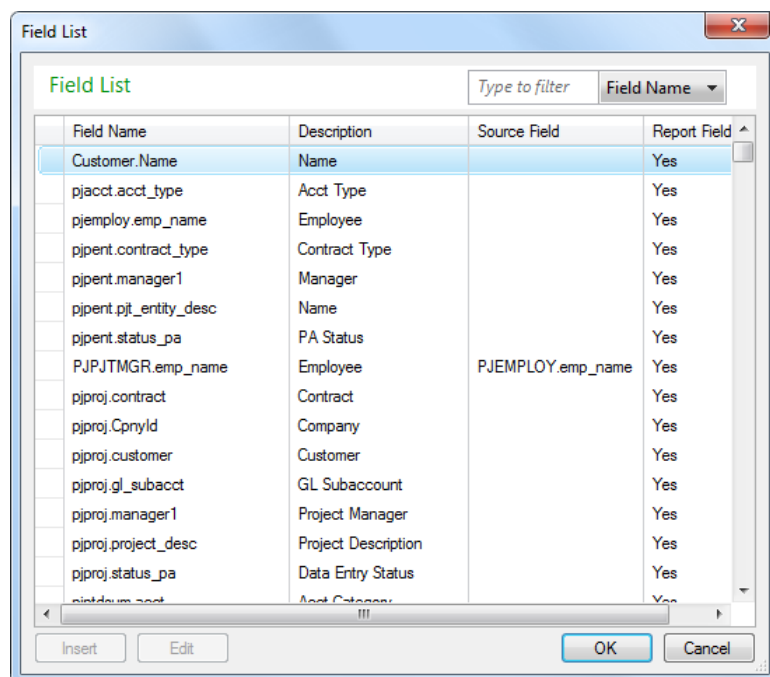


Figure 56: Field List

Following are field descriptions for the *Field List*.

Field Name

Identifies the database record.fieldname to use for selecting report information to be printed.

Description

Description displays the field name of the record. If the field name has been changed using *Field Description Maintenance* (21.405.00), the **Custom Description** will appear here. For more information on changing this description, see “Setting up Field Descriptions” in the Shared Information Help or user’s guide.

Source Field

Source Field will have an entry if the Field Name is from a view in the database, not the actual table.

Report Field

The **Report Field** will display Yes if the report exists on the report you are running. Fields with a No displaying do not print on the report but are available for selection and/or sorting the report.

Report Control Maintenance (98.300.00)

In *Report Control Maintenance (98.300.00)*, you can select many different ROI options for a specific report. At a minimum, at least one report format must be defined for each report number. The report format consists of a description of the format and the name of the Crystal Reports definition file that generates the report format. For information on setting a default report format see “Setting up Default Report Formats” in the Shared Information Help or user’s guide.

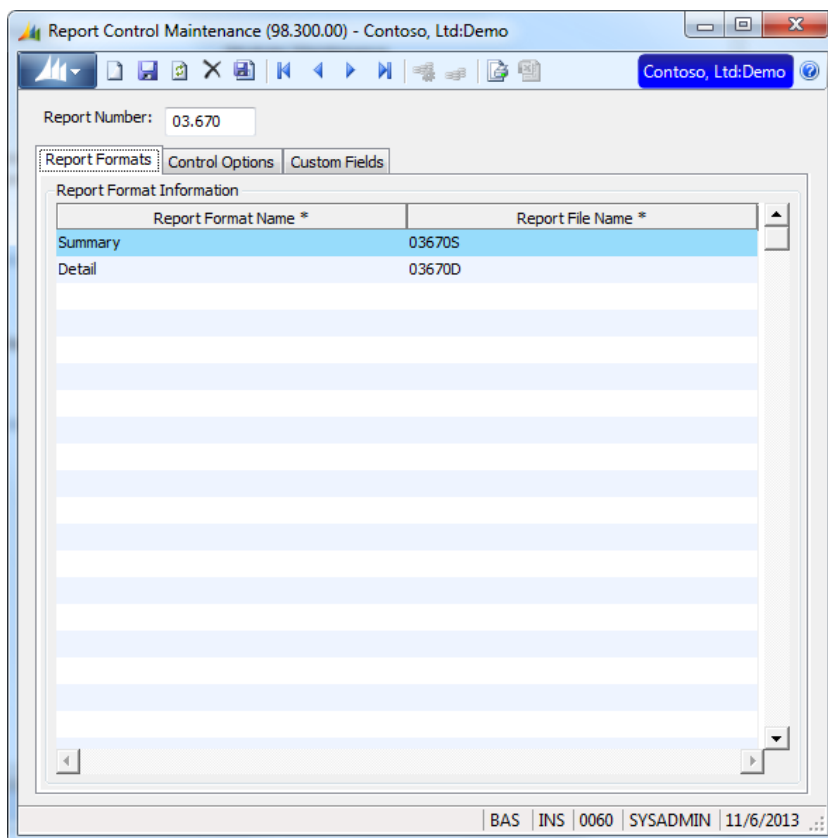


Figure 57: Report Control Maintenance (98.300.00)

Following are field descriptions for *Report Control Maintenance (98.300.00)*.

Report Number

The Microsoft Dynamics SL screen number for the report.

Report Format Tab

An unlimited number of report formats can be added for each report number.

Report Format Name

Report Format Name describes the form of the report. A report may have multiple formats, for example, Detail and Summary.

Report Format

Each report format has its own Crystal Reports file. **Report Format** indicates the name of the file for a report format.

Control Options Tab

Control Options tab is used to identify processing options to be used during the processing of the report.

Figure 58: Control Options tab

Following are field descriptions for **Control Options** tab.

General Reporting Options Area

Report Date Caption

Report Date Caption is used to identify the date field for report date. This field will allow users to change the date used when generating the report.

Pre-Process Name

Pre-Process Name is the name of a Microsoft Dynamics SL Software Development Kit (Microsoft SL SDK) application or stored procedure that will run prior to the generation of the report. This process can be used to prepare any required data or tables for use in the report. For more information, see “Understanding Pre-Processes” on page 37.

Post-Process Name

Post-Process Name is the name of a Microsoft SL SDK application or stored procedure that will run after the generation of the report. This process can be used to remove temporary data or update data based on the completion of the report.

Data Source

Data Source identifies the database type to be used when accessing the data for this report. Application Database indicates that the data will come from the accounting applications. System Database indicates that the data will come from the system.

Disable Banner Prompt (check box)

Disable Banner Prompt, when selected, indicates that the ROI **Cover Page** tab is to be disabled, effectively disallowing the use of a cover page for the report. If this box is not selected, the tab and cover page options are available.

Allow Multiple Copies (check box)

Allow Multiple Copies, when selected, indicates that multiple copies can be created during one generation of the report. If this box is not selected, only one copy of the report will be created when you run the report.

Multi-Company Selection Allowed (check box)

Multi-Company Selection Allowed, when selected, indicates that this report can be generated from data spanning multiple companies. It enables the ROI **Company Selection** tab, where the user can select companies they wish to include on the report. If this box is not selected, report data will be taken from only the company the user is currently logged into.

Transactional Reporting Options Area

Reporting Range Prompt

Reporting Range Prompt determines how reporting dates are handled. Your options are:

- No Period Number – No period or date specified
- Period to Report – To specify a single period
- Beg\End Period to Report – To specify a range of periods
- Calendar Year – To specify a calendar year
- Validated Period to Post – To specify only a period that is not closed
- Date to Report – To specify a single date
- Date Range – To specify a range of dates

Default Period From

Default Period From is enabled if **Reporting Range Prompt** is not No Period Number. Choose the module that will be used for determining the period data.

Period Field Name

Period Field Name is the name of the database field you specify as the period information source. For example, if you want to pull period information from the GLTran table, you could type GLTran.PerPost.

If Date to Report or Date Range is selected from the **Reporting Range Prompt** list, the **Period Field Name** field label changes to **Date Field Name**.

Date Field Name

Date Field Name is the name of the database field you specify as the source for date-specific report information. For example, if you want to pull information from the PRDoc table, you could type PRDoc.ChkDate. The field specified here should have a database type of smalldatetime.

Special Form Reporting Options Area

Print On Special Forms (check box)

Print On Special Forms, when selected, indicates that the report will be printed on special forms.

Display Acct/Sub Fields (check box)

Display Acct/Sub Fields, when selected, indicates that the account and subaccount fields should be displayed and enabled for user interaction on the ROI **Report** tab.

Document Number Caption

Document Number Caption is used to enter text that identifies the document number being displayed in a corresponding control.

Custom Fields Tab

Custom Fields tab is used to specify the values that will be displayed on the ROI **Options** tab. These fields allow a report designer to prompt for values not already displayed in the ROI window.

Figure 59: Custom Fields tab

Following are field descriptions for **Custom Fields** tab.

Custom String Field Captions Area

Custom String Field Captions are used to identify the usage of report-specific text boxes where users can enter values in response to the prompt specified in the caption.

Logic for the use of these values will be in the report itself. These fields are stored in the LongAnsCaptionNN fields in the RptControl record created for a new report or updated for an existing report.

At report generation time the values specified by the user will be in two places: The RiParam LongAnswerNN and the RptRuntime LongAnswerNN fields. For more information, see “RIParams” and “RptRuntime.”

Custom Logical Field Captions Area

The **Custom Logical Field Captions** are used to identify the usage of report-specific check boxes where users can select options based on the prompt specified in the caption.

Logic for the use of these values will be in the report itself. These fields are stored in the LongShortAnsCaptionNN fields in the RptControl record created for a new report or updated for an existing report.

At report generation time the values specified by the user for the available options will be in two places: The RiParam ShortAnswerNN and the RptRuntime ShortAnswerNN field. For more information, see “RIParams” and “RptRuntime.”

Appendix A: Converting Custom Reports to Crystal Reports 2008

All standard Microsoft Dynamics SL reports now use the Crystal Reports 2008 schema. Custom reports your organization created using an earlier version of Crystal Reports should continue to print without the need for modification.

To upgrade your custom reports:

Simply open each report in Crystal Reports 2008, and then save it.

Helpful Tips

Listed below are resolutions to issues you may encounter when you run your custom reports using Crystal Reports 2008.

Issue

The message “Error in file. Unable to connect: incorrect log on parameters” appears when you try to run a custom report.

Solution

Set the database location for the report and any related subreports to point to a new Microsoft Dynamics SL database.

To set the database location:

1. Open the report in Crystal Reports 2008.
2. Select **Database | Set Datasource Location**.
3. Select a new database location. For assistance, click **Help** in the *Set Datasource Location* window.

Issue

The database login window appears when you try to run a custom report.

Solution

Set the database location for the report and any related subreports to point to a new Microsoft Dynamics SL database.

1. Open the report in Crystal Reports 2008.
2. Select **Database | Set Datasource Location**.
3. Select a new database location. For assistance, click **Help** in the *Set Datasource Location* window.

Issue

Some fields are only partially viewable after you convert a report to run in Crystal Reports 2008.

Solution

Make the fields tall enough to display the data.

Issue

The message “Error in formula Record Selection” appears.

Solution

This issue can occur if changes to the RI_WHERE field in the RptRuntime record were made using T-SQL syntax (see “Report Preprocess Tips” on page 138). In the runtime files generated by earlier Crystal Reports versions, it was possible to make corrections in the SQL statement that created a report. However, in Crystal Reports 2008, direct access to the SQL statement is not available at runtime. Instead, the ROI screen modifies the report’s record selection formula. To accomplish this, the contents of the RI_WHERE field must use the same syntax as the report’s record selection formula.

The resolution for this issue depends upon how you implemented the report. In Microsoft Dynamics SL, data coming from the RptControl record, the ROI **Select** tab, and report pre-processes can modify the WHERE clause.

Additional Information

Specifying a Date Filter in the RptControl Record

For a custom report that is date or period specific, you must identify the date field to use when the system searches for report data. Click **Control Options** tab on *Report Control Maintenance* (98.300.00). Then type the name of the date filter in **Date Field Name** or **Period Field Name** by using the format `<Tablename>.<Fieldname>`. This new format is a Crystal Reports 2008 requirement.

Report Preprocess Tips

ROI.exe creates an RptRuntime record before it calls a preprocess. This record can include a field named RI_WHERE that may contain data. The preprocess can append additional criteria to RI_WHERE and update the database.

Because the value in the RI_WHERE field updates the record selection formula when you generate a report, the field must be in the correct format for a record selection formula.

Here are other key changes that *must* be made to preprocesses or when you call ROI directly:

- Enter database fields in the form `{Table.Field}`. Make sure that you use braces as in the example.

Note: In earlier versions of Crystal Reports the braces were not required.

- Date values should be denoted as `Date(<year>, <month>, <day>)` as in the example, `Date(2009, 10, 23)`. This syntax produces an improvement in performance because processing the filter occurs on a computer that runs SQL Server instead of on a workstation.
- Conditional attribute properties are slightly different from those of Transact-SQL. The IN statement for Crystal Reports 2008 is displayed as `IN [<value>,<value>,...]` versus `IN (<value>,<value>,...)` for Transact-SQL.
- For improved report performance, use “ISNULL({Table.Field}) = TRUE” when you select based on a value of NULL.

This is not a complete list of differences between Transact-SQL and selection formula syntax that is acceptable for Crystal Reports 2008. For more information, see the Crystal Report 2008 help and other documentation.

Index

A

Access 58

C

companies, selecting for reports 15
 Control Options 27
 Control Options (98.300.03) 133
 cover page, printing 14, 128
 creating templates 8, 121
 Crystal Reports information 92
 Custom Fields 26, 135

D

data, selecting for a report 12, 124
 default printer
 overriding setup 20
 setting up 3
 defining report settings 6, 119

F

Font dialog box 4

M

maintaining
 report control information 25
 report processing information 27
 report runtime options 26
 menu
 add 55
 Microsoft updates, effect of 63
 multiple companies, printing reports for 15

O

overriding default printer setup 20

P

pre-processes
 Microsoft SL SDK application 38
 SQL Server stored procedure 37
 previewing a report 17
 Print Setup dialog box 3
 printer setup 3, 115
 printing
 cover page 14, 128
 reports for multiple companies 15
 to a file 21
 to a sharepoint site 23

R

report
 defining settings 6, 119
 previewing 17
 printing a cover page 14, 128
 printing for multiple companies 15, 129
 selecting companies to report on 15, 129
 selecting data 12, 124
 setting options 13, 127
 template, saving a 8, 121
 Report Control Maintenance (98.300.00) 25, 132
 Report Option Interpreter (ROI) definition 5, 25
 ROISelect
 for Crystal Reports 51
 for SSRS 51
 formula elements 50
 formula examples 50

S

saving
 report template 8, 121
 SQL statements 63
 selecting
 companies to report on 15
 data for a report 12, 124
 fonts 4
 report options 13, 127
 sort parameters 10, 122
 wildcards 12
 setting up the default printer 3
 sorting data 10, 122
 SSRS Report Viewer 19
 System messages 6202, 6203 63

T

template 121
 creating 8
 using 9

U

using
 templates 9, 121
 wildcards 12

W

wildcards 12