Infor ERP SyteLine

Reporting Guide

version 8.02
Contents

Adding Notes to Reports ................................................. 29
  Adding Notes as a Subreport ........................................... 29
  Adding Notes through a SyteLine Form ............................... 30
Modifying Report Logos .................................................. 31
Preventing Overrides of Date/Number Formats or Fonts ............... 32
Translating Messages from Reports ..................................... 33

Linking Reports to SyteLine ............................................. 35
  Adding a Background Task for the Report ............................ 35
  Creating/Modifying a Form to Run the Report ......................... 36
    Creating a Report Criteria Form .................................. 36
    Add a Field with a Drop-Down List for the Report Input Parameters .......... 36
    Add a Button and a Sequence of Event Handlers to Launch the Report ............ 37
    Testing the Report ................................................. 37
    Adding Date Offset Check Boxes ................................ 37
    Adding Report Preview to the Form ................................ 39
    Modifying the Form to Submit Tasks to the Background Queue ................. 39
  Placing a Task on the Background Queue .............................. 40
  Checking the Status of the Report .................................. 40

Using Infor Framework TaskMan ......................................... 41
  Installation and Configuration ...................................... 42
    Standard Installation .............................................. 42
    Report Developer Installation .................................... 43
    Setup the Utility Server to send E-mail Notifications ................. 43
  Running Stored Procedures, Executables, and Methods Through Infor Framework TaskMan 44
    Stored Procedures .................................................. 44
    Executables ....................................................... 45
    IDO Methods ...................................................... 46
  Infor Framework TaskMan Substitution Keywords .................... 47
  Running Infor Framework TaskMan in Debug Mode .................... 49
    Enable Debug Mode On Process Defaults Form ............................ 49
    Enable Debug Mode By Restarting Service ............................. 49
  Checking Infor Framework TaskMan Events in the Event Log .......... 50
  Using SQL Profiler to Trace TaskMan Instances ...................... 51
  Specifying Report Options On the Process Defaults Form ............... 52
## Troubleshooting

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where to Find Error Message Information</td>
<td>55</td>
</tr>
<tr>
<td>Crystal Reports Are Not Printing</td>
<td>56</td>
</tr>
<tr>
<td>E-mail Notifications Are Not Being Received</td>
<td>57</td>
</tr>
<tr>
<td>Setting Up E-mail Access for Crystal Reports</td>
<td>57</td>
</tr>
<tr>
<td>Report Problems</td>
<td>59</td>
</tr>
<tr>
<td>Missing RPT File</td>
<td>59</td>
</tr>
<tr>
<td>Truncated Data in Text Output</td>
<td>59</td>
</tr>
<tr>
<td>File Not Found</td>
<td>59</td>
</tr>
<tr>
<td>Labels Not Replaced with Strings Table Values</td>
<td>62</td>
</tr>
<tr>
<td>Formatting Problems When Exporting Report Output</td>
<td>63</td>
</tr>
<tr>
<td>Other Infor Framework TaskMan Problems</td>
<td>64</td>
</tr>
<tr>
<td>TaskMan Service Does Not Start</td>
<td>64</td>
</tr>
<tr>
<td>Changes to Intranets Form Settings Are Ignored</td>
<td>64</td>
</tr>
<tr>
<td>No Output</td>
<td>64</td>
</tr>
<tr>
<td>Intermittent Errors</td>
<td>65</td>
</tr>
<tr>
<td>Reports Fail with Return Code -1,073,741,819</td>
<td>65</td>
</tr>
<tr>
<td>Missing TMMsgs.dll</td>
<td>65</td>
</tr>
<tr>
<td>Batches of a Report Fail</td>
<td>65</td>
</tr>
<tr>
<td>Notes Do Not Print on a Report</td>
<td>65</td>
</tr>
<tr>
<td>No Default Set for Parameter n</td>
<td>66</td>
</tr>
<tr>
<td>Missing or Out-of-Date Export DLL</td>
<td>66</td>
</tr>
<tr>
<td>Background Task Runs But Has No History Record</td>
<td>66</td>
</tr>
<tr>
<td>Error 13: Type Mismatch</td>
<td>67</td>
</tr>
<tr>
<td>Error 128: Error Running Report</td>
<td>67</td>
</tr>
<tr>
<td>Error 500: Not Enough Memory for Operation</td>
<td>68</td>
</tr>
<tr>
<td>Error 515: Error Starting Report</td>
<td>68</td>
</tr>
<tr>
<td>Error 534: Error Detected by Database DLL</td>
<td>68</td>
</tr>
<tr>
<td>This Field Name Is Not Known</td>
<td>68</td>
</tr>
<tr>
<td>Empty Columns Appear When Exporting to Excel</td>
<td>68</td>
</tr>
<tr>
<td>ActiveX Component Can’t Create Object</td>
<td>69</td>
</tr>
<tr>
<td>The Transport Failed to connect to the server</td>
<td>69</td>
</tr>
<tr>
<td>Event Messages from Infor Framework TaskMan</td>
<td>70</td>
</tr>
<tr>
<td>Infor Framework TaskMan Debug Mode Messages</td>
<td>73</td>
</tr>
<tr>
<td>RunReport Debug Messages</td>
<td>82</td>
</tr>
<tr>
<td>Sample Debug File</td>
<td>82</td>
</tr>
<tr>
<td>Using the RunReport Debug Files</td>
<td>86</td>
</tr>
</tbody>
</table>
Using RunReport .......................................................... 89
  MGReportProcessor .................................................. 90
RunReport from the Command Line ............................... 90
  Syntax ................................................................. 90
  Switches .............................................................. 90
  Examples .............................................................. 92
Substitution Keywords ................................................. 93
Debug Mode ............................................................. 93

SyteLine Stored Procedure Example ............................. 95

Index ................................................................. 99
Overview

About This Guide

This document describes how SyteLine handles reports. It includes a description of the reporting system architecture, specifics of creating Crystal Reports for SyteLine, how to link custom reports into the SyteLine system, specifics of using the Infior Framework TaskMan and RunReport executables, and how to troubleshoot problems with reports.

For More Information

Additional information about reports, background processing, and customization may be found in the following documents. Most are available from the SyteLine documentation area of the Infor support site (http://www.infor365.com).

- Architectural Guidelines for Customers.
- Infor ERP SyteLine System Administration Guide (Task Manager chapter).
- Infor ERP SyteLine online help files.
- Infor ERP SyteLine Crystal Reports Training Guide. This document is available through the Infor Knowledge Zone (http://www.infor365.com).
- Crystal Reports help files, loaded on the machine where Crystal Reports is installed. The help files are in the Help\En subfolder of the Crystal install folder.

Architecture of SyteLine Reporting

The first sections below explain, from a user’s point of view, how a report is processed for printing or for previewing, and the forms that are used to process it. The following sections explain what is happening behind the scenes during the print or preview.

User View: Printing a Report

Defining the Task

Each report is predefined as a task through the Background Task Definitions form. For reports, the task’s Executable Type is RPT, and the Executable Name is the name of the Crystal Reports RPT file without the RPT extension. Also, Max Concurrent is set to 20,
since Crystal Reports does not allow more than 20 instances of a given report to run at the same time.

Starting the Report
To run a report, a user opens the SyteLine report form and selects the report parameters.

When the user enters the parameters for a report and clicks **Print**, an event attempts to add the task to the **ActiveBGTasks** table, which is essentially the task queue.

Alternatively, the user could schedule the report to run at a specific time or a specific interval by selecting **Actions>Background** from the report form, to display the **Background Queue** form. The scheduled task is then placed in the ActiveBGTasks table with a status of **WAITING**. When the specified time/interval is reached, the status changes to **READY** and the task can be run.

Checking for Excluded Tasks
Before adding a task to the ActiveBGTasks table, the system first checks the list of excluded tasks for this report, set up on the **Excluded Tasks** form. This lists any other
tasks that cannot be active when this report task runs. If there are any excluded tasks listed for this report, the system checks whether any of the listed tasks is currently in the ActiveBGTasks table. If so, the system displays this message: "This task can’t be submitted at this time. Please check the Excluded Tasks table."

Checking the Status of Active Tasks

Information about the task queue is stored in the **ActiveBGTasks** table and can be viewed through the **Active Background Tasks** form. Tasks are marked as WAITING (on hold until a scheduled time), READY (waiting for resources), or RUNNING.

A history record for the task is also created in the **BGTaskHistory** table (see below) when the task is added to ActiveBGTasks.

Executing the Task

The **Infor Framework TaskMan** application, commonly known as TaskMan polls the ActiveBGTasks table, looking for READY tasks. It then executes the tasks when resources are available. TaskMan runs as a Windows service on the utility server.

TaskMan determines that this task is a report by looking at the task’s type in the ActiveBGTasks table, so it routes the task to RunReport.exe. These executables are Infor-written wrappers for the Crystal Decisions API. Once the task is started, TaskMan resets the task’s status in ActiveBGTasks to RUNNING. It also updates the task information in the BGTaskHistory table, which can be viewed through the Background Task History form (see below). You can view information there about active tasks as well as completed ones.

RunReport.exe executes the RPT file, which calls a SQL Server stored procedure in the SyteLine database to retrieve the data needed for the report. The data is then formatted according to the settings in the RPT file.

Options for Printing the Report

If the report is to print in a language other than English, TaskMan searches the Forms database specified in the Sites/Entities form for the Strings table associated with the current SyteLine session. It uses that Strings table to translate the strings used for the report labels.

TaskMan then controls the printing, using either the default system print settings or the settings specified for a certain user/report combination through the Report Options form. (If the Output Format field is set to Printer, the user can enter the name of any network printer; otherwise output goes to the TaskMan server’s default printer.)

The printing process may also take into consideration fonts, language, and document profiles, as described on pages 18 through 20.

After the Task Is Completed

When the task completes, TaskMan deletes the record from the ActiveBGTasks table. The following task information is updated in the BGTaskHistory table and Background Task History form:
Overview

- Task parameters passed from SyteLine to TaskMan, set when the report is submitted
- Start time, set when TaskMan starts the report
- Stop time, set when the task finishes
- Success/failure indicator, set when the task finishes
- Return codes, set when the task finishes.

A return status of 0 indicates that the process completed successfully. Error messages are routed to the appropriate logs.

User View: Previewing a Report

First, the report must have been predefined as a task through the Background Task Definitions form (as described on page 7).

Starting the Preview

To preview a report, a user opens the SyteLine report form and selects the report parameters.

When the user enters the parameters for a report and clicks Preview, the preview task is added to the ActiveBGTasks table, similar to the way report printing is handled. After the report preview file is built, a notice that the file exists is placed on a queue. The
PrintPreview application polls the queue and, when it sees the notice, finds the report preview file and displays it.

Options for Previewing the Report

If the report is to preview in a language other than English, TaskMan searches the Forms database specified in the Sites/Entities form for the Strings table associated with the current SyteLine session. It then uses that Strings table to translate the strings used for the report labels.

The output format for report previews is defined on the Intranets form. HTML is the default preview format. For more information about the Intranets form, see the Infor ERP SyteLine online help.
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WinStudio submits a report print task through <code>BGTaskSubmitSp</code>, which inserts a task request record into the <code>ActiveBGTasks</code> table.</td>
</tr>
<tr>
<td>2</td>
<td>TaskMan service's polling thread monitors the <code>ActiveBGTasks</code> table through IDO runtime, looking for records with Ready status, which tells TaskMan to pick up and process the record.</td>
</tr>
<tr>
<td>3</td>
<td>While processing the requested report task, TaskMan creates a parameter file in <code>\&lt;ABC&gt;\Report\ParmFiles</code>. The file name is XML: <code>ParmFile_&lt;TaskExecutable&gt;_&lt;Site&gt;_&lt;TaskNumber&gt;.xml</code>, and is generated through a class library <code>MGReportProcessor.dll</code>. This class library provides report parameter type (<code>ReportTaskParms</code>), allowing the caller to persist/parses parameters needed to execute reports (simply using the &quot;XmlSerializer&quot;). Please see a sample XML in [Report Parameter XML File - Example], shown at the end of message. <code>MGReportProcessor</code> deletes the XML parameter file after successfully executing the report; if the task fails or if you are using Debug mode, the file will not be deleted so that we can easily rerun and debug the report tasks (simply executing &quot;RunReport&quot; with its -parmfile &quot;&lt;ParmFile&gt;.xml&quot;).</td>
</tr>
<tr>
<td>4 - 5</td>
<td>After creating a parameter XML file, TaskMan executes <code>RunReport</code>, which causes <code>MGReportProcessor</code> to load the specified parameter XML file.</td>
</tr>
<tr>
<td>6</td>
<td><code>MGReportProcessor</code> loads the requested Crystal Report RPT file, first looking in <code>\&lt;ABC&gt;\Report\Reports\&lt;Config&gt;</code>. This is a report specific to the current configuration. If not found, then it loads the rpt file found at the default location of <code>\&lt;ABC&gt;\Report\Reports</code>.</td>
</tr>
<tr>
<td>7</td>
<td><code>MGReportProcessor</code> uses the IDO runtime to query string objects and update <code>BGTaskHistory.ReportOutputPath</code> after successfully executing the report.</td>
</tr>
<tr>
<td>8</td>
<td>Default report output path is: <code>\&lt;ABC&gt;\Report\OutputFiles\&lt;UserId&gt;\&lt;ReportFileName&gt;_&lt;Site&gt;_&lt;TaskNumber&gt;.&lt;FileExtension&gt;</code> For more information, see “Determining the Output Path” on page 15.</td>
</tr>
<tr>
<td>9</td>
<td>As successfully exporting the requested report, <code>MGReportProcessor</code> updates <code>BGTaskHistory</code> with the actual output file path (<code>BGTaskHistory.ReportOutputPath</code>).</td>
</tr>
</tbody>
</table>
New Developer View: Previewing a Report
Determining the Output Path

There are several ways of defining your report output. Whatever the output directory ends up being, TaskMan looks for a user-specific subfolder at that location. If the subfolder is not found, TaskMan appends that subfolder to the output path, except in the case of Report...
Output Obfuscation. For more information about these options, see the SyteLine online help.

Report Options

The Report Output Directory field, on the Report Options form, allows you to specify your own output location for each user/report task combination. You can specify a valid file system/UNC path (e.g. \XYZ\ReportOutputs), and report outputs will be generated at that location. Entering an output directory here will override any path defined on the Intranets or Process Defaults forms.

Process Defaults

The Default Name field, on the Process Defaults form, allows you to define a default output directory that will override the Task Man Path field on the Intranets form. However, any user/task specific output directory defined on the Report Options form will override this default setting. If you have overridden the TaskMan Path by specifying a Report Output Directory on the Process Defaults form, then use the Report URL field to specify the URL path to that directory.

Report Obfuscation

Report Obfuscation allows you to distinguish output by the session ID appended to the report name, rather than by a user subfolder. This option is useful if you want to expose the report output directory over the Web without indexing on the folder, so that users cannot easily access the reports of other users.

Report URL

If you specify a URL in the Report URL field (on the Intranets form), this path will be used in place of the Task Man Path field.
Objects Used in Reporting

Executables

<table>
<thead>
<tr>
<th>Executable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taskman.exe</td>
<td>Runs as a service on the utility server. It monitors the queue of tasks. See Chapter 4, &quot;Using Infor Framework TaskMan.&quot; It references MGReportProcessor.dll to generate the task parameter (XML) file, which is passed to RunReport as its command-line input parameter</td>
</tr>
<tr>
<td>RunReport.exe</td>
<td>Wrapper for both the Crystal Decisions API and MGReportProcessor.dll. Can also be run from a command line. See Chapter 6, &quot;Using RunReport.&quot;</td>
</tr>
</tbody>
</table>

DLLs

<table>
<thead>
<tr>
<th>DLL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMMsgs.dll</td>
<td>Used by TaskMan.</td>
</tr>
<tr>
<td>MGCore.dll</td>
<td>Used by Report Preview.</td>
</tr>
<tr>
<td>MGReportProcessor.dll</td>
<td>A .NET class library which contains the core reporting logic, which both TaskMan.exe and RunReport.exe reference.</td>
</tr>
</tbody>
</table>

Tables/IDOs/Forms

<table>
<thead>
<tr>
<th>Table</th>
<th>IDO</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveBGTasks</td>
<td>MGCore.ActiveBGTasks</td>
<td>Active Background Tasks</td>
<td>Queue of tasks.</td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>Background Queue</td>
<td>Schedules tasks to run later.</td>
</tr>
<tr>
<td>BGTaskHistory</td>
<td>MGCore.BGTaskHistories</td>
<td>Background Task History</td>
<td>Is created by a trigger on ActiveBGTasks, updated by TaskMan. Contains a record of when a task was submitted, started and completed, plus any error messages.</td>
</tr>
<tr>
<td>BGTaskDefinitions</td>
<td>MGCore.BGTaskDefinitions</td>
<td>Background Task Definitions</td>
<td>Creates a record that identifies each background task to TaskMan.</td>
</tr>
<tr>
<td>ReportOptions</td>
<td>MGCore.ReportOptions</td>
<td>Report Options</td>
<td>Sets the output format for reports, and the UNC printer name if the report is sent directly to a printer.</td>
</tr>
</tbody>
</table>
Infor ERP SyteLine Reporting Guide
Copyright © 2010 Infor

Stored Procedures/COM Methods

<table>
<thead>
<tr>
<th>SP/COM Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGTaskSubmit</td>
<td>Used with Infor ERP SyteLine to submit reports to TaskMan. It returns the row pointer of BGTaskHistory (@TaskHistoryRowPointer) for a task being created.</td>
</tr>
<tr>
<td>SL.SLJobQueues.BackGroundQueueSP</td>
<td>Used with Background Queue form</td>
</tr>
<tr>
<td>MGCore.ActiveBGTasks.BackGroundQueueDeleteSP</td>
<td>Used with Background Queue form</td>
</tr>
<tr>
<td>CLM_GetBGTasksToProcessSp</td>
<td>TaskMan calls it at its polling time, querying the “ActiveBGTasks” table to retrieve background tasks to be processed.</td>
</tr>
<tr>
<td>GetTaskOptionsSp</td>
<td>Used to determine the report output directory, based on the Report Output Directory field on the Process Defaults and/or Report Options forms.</td>
</tr>
<tr>
<td>UpdateActiveBGTaskSp</td>
<td>While processing requested task, TaskMan is to update the task status using the stored procedure.</td>
</tr>
</tbody>
</table>

Customization vs. Modification of Existing Reports

Changes to existing SyteLine reports can combine customizations and modifications. (The differences between customization and modification, and the guidelines for both, are defined in the Architectural Guidelines for Customers document.) One component of a report is its front-end form, which can be customized like other forms, by using standard Infor ERP SyteLine techniques.

Changes to the other report components - the Crystal Reports report definition file (.RPT), the SQL Server stored procedure that the report is based on, and a background task definition stored in the application database BGTaskDefinitions table - are considered modifications. When a base report needs to be modified, each of these elements should be copied and renamed, and the modifications applied to the copy.

If you need to produce multiple variations of a base report, consider building a version of the report that presents users with all the options they may need as components on the report form, corresponding to parameters passed to the .RPT file and the stored procedure.

Fonts Used in Reports

A report prints in the font assigned to the language in which the report is generated. System administrators can assign a different font to a language in the Language IDs form:

Font names in the Font Descriptor drop-down list represent fonts installed on the SyteLine client machine. To ensure that reports print in the selected font, the font must also be
Barcode type fonts do not need to be installed on the client workstation if they are not going to be previewed there.

The font setting does not apply to barcodes or currency amounts in checks. To change fonts for these objects, edit the objects in the report’s .RPT file in Crystal Reports.

Date and Numeric Formats Used in Reports

When a user prints or previews a report, by default the report uses the date and numeric formats that were defined for the fields in the .RPT file. In the SyteLine Language IDs form, system administrators can assign a different date or numeric format to a language. When that language is selected for a SyteLine session, all reports use the language’s date and numeric formats, overriding the default formats.

However, report designers can specify that certain default date or numeric fields cannot be overridden by adding a prefix to the date or numeric field as described on page 32.

Languages Used in Reports

Some reports print in a language other than the language currently in use on the client workstation. For example, if a language is specified for a vendor in the Vendors form, certain reports, such as the Purchase Order Report, print in the language specified for the vendor. The report uses the font associated with that language in the Language IDs form.

The following reports print in the vendor’s or customer’s language, if specified:

- ARInvoiceCreditDebitMemoReport
- Bill of Lading (prints in the Site’s language)
- ChangeOrderReport
- CustomerStatementsReport
- EstimateResponseFormReport
- OrderInvoicingCreditMemo
- OrderVerificationReport
- PackingSlip
- PostProjectInvoiceMilestones
- PriceAdjustmentInvoice
- ProFormalInvoice (prints in the Site’s language)
- ProjectPackingSlip
- PurchaseOrderReport
- ReprintPackingSlipReport
- ReprintProjectPackingSlipReport
RMACreditMemo

When these reports are printed, TaskMan selects the proper Strings table to use for report labels, based on the language code setting in the Vendors or Customers form. If there is no language code setting for the vendor/customer, the SyteLine user’s language code will be used to print the report. Special text (for example, the CO text on the order invoice) can be translated using the Multi Lingual Order Invoice or Multi Lingual Purchase Orders form.

All other reports print in the SyteLine user’s language. The user language is set at login by using the Windows Control Panel setting. After logging into SyteLine, the user can manually choose a different Strings table by selecting Options>Change String Table. The currently active Strings table will be used to print the report.

Print Preview always displays the report in the user’s language.

Document Profiles

The Customer Document Profile and Vendor Document Profile forms enable you to send certain reports to customers and vendors. When defining the report parameters, select the Use Profile field to trigger use of the document profiles when you print the report (document profiles are ignored when previewing).

The Customer and Vendor Document Profiles forms are subject to record collection caps. For example, if your collection cap is set at 100, and you attempt to send a report to 150 customers, only the first 100 will be processed. For more information, see the About Caps on Collection and Drop-Down Lists topic in the online help.
The following flowchart shows how report output is sent using the Document Profile forms.

Splitting Up of Tasks

All reports that use the Language Code and/or the Customer/Vendor Doc Profile will be split up into smaller tasks prior to submitting to TaskMan. Active Background Tasks and Background Task History will show multiple entries based upon the split.

Using E-mail with Reports

E-mail is used with report in two ways. With both methods, you will need to set the e-mail preferences on the Intranets form before any e-mails can be sent.

Send E-mail to Yourself

The Report Options form allows you to send yourself e-mail notification when a report you submitted is complete. You can also attach a copy of the report to the e-mail notification. On the Intranets form, choose the protocol with which to send the e-mail. For more
Send E-mail to Customers and Vendors

The Customer Document Profile and Vendor Document Profile forms allow you to send a copy of the report to all customers or vendors from whom you have created, and activated, a document profile.

The following flowchart describes the steps needed to email reports to customers or vendors:

- Report is submitted
- Was the Use Profile field selected on the report form?
- Yes
- Does an active customer/vendor document profile exist?
- Yes
- Is the report format defined on the Report Options form?
- No
- Report is processed using the default settings on the Intranets form, and no e-mail is sent to the customer/vendor.
- Yes
- Is the document profile method E-mail?
- No
- Report is sent to document profile recipient using the method and destination defined in the document profile.
- Yes
- Is output format defined on the Report Options form?
- No
- Report is processed using the default settings on the Report Options form, and no e-mail is sent to the customer/vendor.
- Yes
- Is output format defined on the Intranets form?
- No
- Report is e-mailed to the destination in the document profile in the default Crystal Reports format.
- Yes
- Report is e-mailed to the destination in the document profile in the format defined on the Report Options form.
Creating or Modifying Reports

This chapter includes only information specific to creating or modifying a report in a SyteLine environment. It does not include everything you need to know to create or modify a report. Please refer to the Crystal Reports help files for complete details.

Stored Procedures

Structure

All SyteLine report stored procedures have the same basic structure, described below.

- There is a set of input parameters typically defaulted to NULL. A NULL input parameter indicates that the parameter should not be used to restrict the record set returned by the stored procedure. The stored procedure may have additional parameters for notes and the report header.
- There is a block of code setting current session values, starting a transaction, and setting the SyteLine session.
- NULL input parameters used to specify ranges are set to minimum and maximum values. Use the SQL user-defined functions for this: LowDate, LowInt, LowString, HighDate, HighInt, HighString, and so on. There is also the function ExpandKyByType for parameters of type NUMSORTCHAR.
- Date offset parameters are set. (See “Date Offset Parameters” below.)
- Many report stored procedures use temp tables for the report's record set. These should be created by selecting variables into a temp table, so that User Defined Types can be used.
- There is a large block of code populating the result set.
- At the end of the stored procedure, the report record set is returned, the transaction is committed, and the SyteLine session is closed.

Example of a Report Stored Procedure

Appendix A, “SyteLine Stored Procedure Example,” contains a script you can use to add a sample stored procedure to your demo application database. This sample stored procedure (Rpt_ReportDemoSp) is referred to in examples later in this chapter and in the following chapter.
Date Offset Parameters

SyteLine allows reports to be run on a schedule. For example, a report could run every night on the background queue. When run on the background queue, some reports need to automatically increment date ranges. To handle this, you must add a date offset parameter for each date parameter, and then call ApplyDateOffsetSp for each date and offset parameter.

The ApplyDateOffsetSp procedure also handles the use of null date parameters to indicate that all records are to be returned. If the Offset is null, then the date parameter is used. If the offset is not null, then it is added to the current date and used in place of the date parameter's value.

In the Create Procedure section of the report’s stored procedure, add the extra parameters at the end of the parameter list, for example:

```sql
@OrderDateStartingOffset DateOffsetType = NULL,
@OrderDateEndingOffset DateOffsetType = NULL
```

At the top of the report, add lines similar to the following examples for each date field:

<table>
<thead>
<tr>
<th>If the Date Is...</th>
<th>Add a Line Similar to This</th>
</tr>
</thead>
<tbody>
<tr>
<td>a starting date</td>
<td>EXEC ApplyDateOffsetSp @OrderDateStarting OUT, @OrderDateStartingOffset, 0</td>
</tr>
<tr>
<td>an ending date</td>
<td>EXEC ApplyDateOffsetSp @OrderDateEnding OUT, @OrderDateEndingOffset, 1</td>
</tr>
<tr>
<td>current date</td>
<td>EXEC ApplyDateOffsetSp @OtherDate OUT, @Offset If @Offset is NULL, then use the Site Date</td>
</tr>
</tbody>
</table>

These Date Offset parameters are also set up as fields in the report form (see “Adding Date Offset Check Boxes” on page 37).

After setting all of this up, use Crystal Reports Designer to re-synch the method.
RPT Files

Where to Save RPT Files

When you create or modify a report, save the .RPT file in the TaskMan\Report\Reports subfolder on the machine where TaskMan is installed (usually the utility server).

Creating a Report Based on a Stored Procedure

The following example process creates an RPT file based on the sample stored procedure Rpt_ReportDemoSp (see Appendix A, “SyteLine Stored Procedure Example.”)

1. In the Crystal Report Designer, select New, and then click OK to use the Report Expert.
2. Select Create New Connection.
3. Select OLE DB (ADO).
4. Select Microsoft OLE DB Provider for SQL Server.
5. Fill in database connection information and click Finish.
6. Expand the database, if necessary, to see the list of objects.
7. Expand the list of stored procedures. Select Rpt_ReportDemoSp and add it to the list in the right pane.
8. Click OK to default the Stored Procedure parameters to NULL.
9. Click Next, and add all of the fields to the report.
10. Click Finish.
11. Save the report. Typically, reports are named the same as the stored procedure, without the "Rpt_" and "Sp". So our example should be named ReportDemo.RPT.

Verifying the Database Name

Whenever you make any sort of change to a report’s stored procedure, you should open the RPT file in Crystal Designer and select Database>Verify Database. Crystal will then fix the RPT file and make sure everything is up-to-date.

Then select Database>Verify Database again. Crystal should display a message saying that the report is up-to-date. If it does not, make sure the report has been converted to OLE DB (ADO), and make sure the stored procedure does not include any of the problems described under “RPT Files” on page 25.

If you do not verify the database, you may get an "Error detected by Database DLL" message when the report runs through TaskMan.

String Translation

In SyteLine, the text used for labels and list pair display values comes from the Strings table in the Forms database.

Creating Formulas for Labels

In order to get labels from the Strings table, formulas are used for the labels. For a formula to be used as a translatable label, the formula text must include the name of a string in the
Strings table in parentheses and double quotes: ("StringName"). Before running a report, TaskMan loops through it, modifying formulas that include labels. It uses the quoted string name to look up a string in the Strings table. TaskMan then replaces the string name in the formula text with this new string. TaskMan also replaces any occurrences of "\n" with "Chr(13) + Chr(10)", the VB command for a new line.

The following example shows how you would set up a report label:

1. Open the ReportDemo.RPT report you created on page 25.
2. Delete the CoNum text field.
3. If the Field Explorer window is not visible on the right, select View>Field Explorer. Then select Formula Fields, right-click and select New.
4. Enter sOrder as the formula name.
5. Enter ("sOrder") as the formula text.
6. Save and close the formula editor.
7. Drag and drop the @sOrder formula to where the CoNum field used to be. You may have to slide the other fields to the right to make room.

This process would have to be repeated for the other labels.

Modify the Report to Use the List Pair Display Values

Our sample report’s status contains an example of a field that uses a SyteLine list pair. List pairs are used in fields where the value displayed on the SyteLine form is different than the value stored in the database. For example, the type field in the database can hold values "C", "H", "O", "P", "Q", "S", or "W", but the values displayed in the forms are "Complete", "History", "Ordered", "Planned", "Quoted", "Stopped", or "Working". The display values are stored in the Strings table, where the string name is 's' + property_class_name + '=' + field_database_value. For example, the type field is of property class CoNum, and for type "C", the string name is "sCoStatus=C" and the string value is "Complete". Crystal Reports formulas are used to get the List Pair display values.

The following example shows how you would set up a formula for a list pair:

1. In Crystal Reports Designer, open the ReportDemo.RPT report you created on page 25.
2. If the Field Explorer window is not visible on the right, select View>Field Explorer. Then select Formula Fields, right-click and select New.
3. Give the formula the name Status.
4. For the formula text, enter:

```vbnet
if {Rpt_ReportDemoSp;1.Status} = "C" then ("sCoStatus=C")
else if {Rpt_ReportDemoSp;1.Status} = "H" then ("sCoStatus=H")
else if {Rpt_ReportDemoSp;1.Status} = "O" then ("sCoStatus=O")
else if {Rpt_ReportDemoSp;1.Status} = "P" then ("sCoStatus=P")
else if {Rpt_ReportDemoSp;1.Status} = "Q" then ("sCoStatus=Q")
else if {Rpt_ReportDemoSp;1.Status} = "S" then ("sCoStatus=Q")
else if {Rpt_ReportDemoSp;1.Status} = "W" then ("sCoStatus=W")
else ""
```

5. Save and close the dialog.
6. Drag and drop the formula onto the report.
If your objects database has the SyteLineLabelStringIDs, you can get the list pair values with the following select statement. Replace `SyteLine_Forms` and `SyteLine_Objects` with the appropriate database names:

```sql
select s.Name, s.String, l.SqlTableName, l.SqlColumnName
from SyteLine_Forms.dbo.Strings s,
     SyteLine_Objects.dbo.SyteLineLabelStringIDs l
where s.Name like 's' + l.FormatDomain + '%$'
    and l.SqlTableName = 'co'
    and l.SqlColumnName = 'type'
```

Adding Barcodes to a Report

The system implements barcoding by loading the barcodes as fonts on the TaskMan machine, then setting the report field to use the barcode font. To add a barcode:

1. Create a formula field
2. Place asterisks on either side of the string that you want barcoded.
3. Place this field on the report.
4. Select a barcode font (for example, Code39QuarterInch).
5. Set the font size.

**CAUTION:** Field fonts are typically reset at runtime. See “Preventing Overrides of Date/Number Formats or Fonts” on page 32 to learn how to prevent your barcode fonts from being reset.

Resynchronizing a Report After Modifying a Stored Procedure

To resynchronize the RPT file with the changed stored procedure:

1. Open the report in Crystal Report Designer.
2. Select **Database>Set Datasource Location**.
3. In the Replace With pane:
   a. Expand **Create New Connection**.
   b. Select **OLE DB (ADO)**.
   c. Select **Microsoft OLE DB Provider for SQL Server**.
   d. Fill in database connection information and click **Finish**.
   e. Scroll down in the pane and expand the database name.
   f. Expand **Stored Procedures**.
   g. Scroll down and highlight **Rpt_ReportSp;1** where `Report` is the report name.
   h. Click **Update**; then click **Close**.

   This displays the Enter Parameters dialog where you choose stored procedure parameters.

4. In the Enter Parameters dialog, enter a logical value for each parameter; NULLs may work for everything. Click **OK**.
5. If you changed parameters, the following messages will appear. Click OK after each of them:
   "The database file Rpt_ReportSp;1 has changed. Proceeding to fix up the report!"
   "The database is now up to date."

6. If a dialog titled "OLE DB (ADO)...Choose a data source from the list" appears, click Cancel.

7. Click Save.

8. Scroll down to and double-click on ReportNotes.rpt if it appears; if not, skip to Step 13.

9. Select Database>Set Location.

10. In the Current Data Source box, highlight ReportNotesView.

11. In the Replace With pane:
   i. Your data source name/database name should still be expanded. If not, expand as described above.
   j. Collapse Stored Procedures.
   k. Expand Views.
   l. Scroll down and highlight ReportNotesView and click Update.
   m. Click Close.

12. Select Database>Verify Database. The message "The database is now up to date" should appear. Click OK.

13. Save the report.

14. In the related SyteLine form, modify the parameter list passed to Run Background Task to include any new parameters from the stored procedure.

15. Test.
Adding Notes to Reports

The toolset supports several different kinds of notes. The ones that have to be added to reports are the System and Specific Object Notes. Object notes are associated with a particular record in a table, so to get an object note you need the record’s RowPointer. Notes can be marked for internal use only, or for external distribution (for example, on an invoice). Users choose whether to print Internal Notes and/or External Notes when they print a report.

The notes that have to be added to reports come from four tables:

- ObjectNotes
- NoteHeaders
- SpecificNotes
- SystemNotes.

There is a view in the application database, ReportNotesView, that joins these tables together. It contains three fields relevant to reports:

- Note - contains the text of the note.
- RefRowPointer - is used to join the view to the appropriate table and record. (TableName is no longer needed to do this.)
- IsInternalNote - indicates whether the note is for internal or external use.

To retrieve the notes for a report, ReportNotesView must be joined to one of the tables in the report. This can be done in the stored procedure or by adding a subreport to the report. Adding it to the stored procedure is more efficient; however, adding a subreport to the Crystal RPT file is an easier report modification.

Adding Notes as a Subreport

The report ReportNotes.rpt can be added as a subreport to any existing report to display notes. The steps below add a note to the sample ReportDemo report.

1. Modify the main report’s stored procedure by adding two new input parameters, @ShowInternal and @ShowExternal. For example:

   ```sql
   ALTER PROCEDURE <report_name> (  
     @StartItem  ItemType = null,  
     @EndItem    ItemType = null,  
     @ShowInternal FlagNyType = 1,  
     @ShowExternal FlagNyType = 1  
   ) AS  
   BEGIN  
   ...
   ```

   Replace <report_name> with the name of the report. (The sample ReportDemo script in Appendix A, “SyteLine Stored Procedure Example” includes these parameters.)

2. Add RowPointer and a call to the function `dbo.ReportNotesExist` to the columns returned by the report’s stored procedure.
SELECT it.item, it.description,
       it.RowPointer,
       NoteExists = dbo.ReportNotesExist('item', it.RowPointer, @ShowInternal,
                                    @ShowExternal, it.NoteExistsFlag)
FROM Item it

In steps 2 and 4, the purpose of the NoteExists column is to allow a report to be set up so that it only tries to retrieve notes if at least one note exists. This can be omitted and the report will still work. It will just be less efficient.

3. Open the sample ReportDemo report in Crystal Designer and select Database>Verify Database.
4. Insert a new section to contain the subreport by selecting Format>Section, highlighting the appropriate section and clicking Insert. The sample report added Details section b. Create a formula to conditionally suppress this section by clicking the x+2 button next to the Suppress check box.
5. Enter a formula similar to this:
   {DemoReportSp;1.NoteExists} = 0
   We recommend that you select from the list of report fields, to prevent typing errors.
6. Add the subreport by selecting Insert>Subreport. Then:
   a. Select the Choose a report radio button and add ReportNotes.rpt.
   b. Edit out the path to the subreport, if any.
   c. On the Link tab, add RowPointer to the fields to link to, and select RefRowPointer from the combo box beneath the "Select data in subreport based on field" check box. (If the combo box is grayed out, check this check box first.)
   d. Add ?ShowInternal to the fields to link to. Make sure the "Select data in subreport based on field" check box is now unchecked. In the "Subreport parameter field to use" combo box, enter ?ShowInternal.
   e. Add ?ShowExternal to the fields to link to. In the "Subreport parameter field to use" combo box, enter ?ShowExternal.
   f. Click OK and drop the subreport onto the main report.
7. You can modify the subreport links by selecting Edit>Subreport Links.
8. To get rid of the box around the subreport, right-click on the subreport and select Border and Colors. On the border tab, change all four Line Style fields to None.

Adding Notes through a SyteLine Form

To add notes through SyteLine, you need to open a form that is based on an IDO whose primary base table is the table to which you want to add a note. In the following query, replace SyteLine_Forms with the name of your forms database, and replace the two occurrences of "item" with any table name to get a list of forms you can use to add notes for that table.
select f.Name, f.PrimaryDataSource, "V(fds_DataSource)" = v.value
from SyteLine_Forms.dbo.Forms f
  left outer join SyteLine_Forms.dbo.Variables v
    on f.PrimaryDataSource = 'V(fds_DataSource)'
    and v.Name = 'fds_DataSource'
    and v.Value like '%item%'
    and f.ID = v.FormID
where f.PrimaryDataSource like '%item%'
  or (f.PrimaryDataSource = 'V(fds_DataSource)' and v.Name IS NOT NULL)

Once you have the list, select one of the forms and open it in SyteLine. To create notes in
the form:

1. Refresh the form and select the record to which you want to attach a note.
2. Select Actions>Notes for Current.
3. Click the New button.
4. Enter a subject, and in the large multi-line edit box, enter the text of the note.
5. Click OK.

Modifying Report Logos

Each report contains a color logo in the upper right corner. If you want to replace that logo
with one of your choosing, ensure that it adheres to the following rules:

- The format must be bitmap (.bmp).
- The size must be 2.00 inches wide by 0.65 inches high to preserve proper report
  formatting.
- The file must be named SLHeaderLogo.bmp.
- The file must be placed on the TaskMan machine, in the ..\Report\Reports directory.
Preventing Overrides of Date/Number Formats or Fonts

If you want a specific date field, numeric field, or font on a report to always display in the default format that you define in the report, use a field prefix of Lflags_, where flags is any combination of the following:

- D - Do not override the default date format
- N - Do not override the default numeric format
- F - Do not override the default font

For example, to ensure that the "last paid" date always displays in the default format and font on a customer sales report:

1. In Crystal Reports Designer, open the .RPT file.
2. Right-click the "last paid" field in the Design pane and select Format Field….
   - In the Common tab, the Object Name property is set to LastPaid1. In the Date and Time tab, this field is formatted as 3/1/99 and the font is set to Arial Regular 10.
3. To prevent user sessions from overriding this format and font, add a prefix to the field’s Object Name property, like this: LDF_LastPaid1.
Translating Messages from Reports

Usually, your custom RPT file should use string formulas for messages, so they will be translated properly. The stored procedure should return an indication that a message is needed, instead of the actual message, and then the report should include selection criteria that determine which message string to display.

However, in some cases it may be appropriate to call MsgAppSp to provide messages. Be aware that, because of the way report stored procedures are initialized, MsgAppSp always uses the session's default language (en-US) for messages displayed by report stored procedures. To work around this, you will need to get the variable for the proper language as described below.

TaskMan creates a session before generating a report. The session variable MessageLanguage contains the language ID of the user requesting the report. TaskMan then passes the Session ID to RunReport as a command line parameter.

1. If a report stored procedure calls MsgAppSp, add this parameter to the stored procedure:
   
   ```
   @BGSessionId nvarchar(255) = NULL
   ```
   
   When RunReport sees this parameter, it passes the Session ID to the stored procedure.

2. In the stored procedure, immediately after the call to InitSessionContextSp, add the following code:

   ```sql
   -- Copy the session variables referenced by @BGSessionId to the current session created by the call to InitSessionContextSp.
   EXEC CopySessionVariablesSp
   @SessionId = @BGSessionId
   ```

   This copies the MessageLanguage session variable to the current session context, so when MsgAppSp is called, it will use the correct language ID.

3. Perform a "Verify Database" on the associated report after the stored procedure has been updated.
Once you have created the Crystal RPT file and the stored procedure for your report, you need to make the report available to users in SyteLine. This chapter describes the steps needed to do that. The steps are organized so that they continue setting up the example report used in the previous chapter.

Adding a Background Task for the Report

Use the Background Task Definitions form to create a background task record for your report.

1. In SyteLine, open the Background Task Definitions form.
2. Add a new task on this form:

   - Task Name - For simplicity, enter the same name you entered for Executable Name.
   - Executable Name - Enter the name of the Crystal RPT file, without the .rpt extension. Continuing our example from the previous chapter, this name would be ReportDemo. The report definition (.RPT file) for this report must be placed in the TaskMan\Reports folder on the server where TaskMan and Crystal Reports are installed.
   - Executable Type - Select RPT.
   - Max Concurrent - Set this to 20 for any report task. Crystal does not allow more than 20 instances of a given report to run at the same time. TaskMan uses the Max Concurrent field to set this cap.
Isolation Level - The setting **Read Uncommitted** allows faster query performance. See the online help for this form and the Process Defaults form for the effects of **Read Uncommitted** versus **Read Committed**.

3. Save the record.

Creating/Modifying a Form to Run the Report

Most reports are run from Criteria forms. These forms typically have several edit boxes for the report’s parameters, and a button to submit the report.

Creating a Report Criteria Form

1. To create a form where users can run the new report:
2. Start SyteLine (Developer Client version) and select **Form>Definition>New**.
3. Enter a form name. This should be the report name with "Report" tacked onto the end (continuing our example, ReportDemoReport).
4. Select **Build From Scratch** from the Form Type drop-down list.

You do not need to select a data source for this form, since it is simply a report options form that passes parameters to the stored procedure.

5. Click **Next** and then **Finish**.
6. In the Form Properties dialog, enter a caption and click the **Caption** button to create it. The caption should be a string that begins with "f".
7. Click **OK** to save the caption, and then click **OK** on the Form properties window.
8. Save the form.

Add a Field with a Drop-Down List for the Report Input Parameters

1. In the new form, select **Edit>Component>New>Enhanced Combo**.
2. Use the mouse to draw a rectangle on the form, or just click in the form to create the component, and resize it later.
3. Give the component a name (for example, OrderStarting).
4. For the Data Source, create a variable with the name OrderStarting and CoNumVar as the component class:
   a. Select the Data tab and click the **Binding** button.
   b. Select **Variable as Type**.
   c. Click **Edit** and enter **OrderStarting** in the Variable edit box.
   d. Click **OK** until you return to the form.
5. Do a right-click copy and paste on this component. Change the component name and variable to OrderEnding.
6. Add date combo fields for OrderDateStarting and OrderDateEnding. Specify OrderDateStarting and OrderDateEnding as the variables used as data sources.

Do not worry about the Date Offset, Note, and Header parameters for now. They will be set up later.
All data fields should be bound to variables with the same name as the data, plus "Starting" or "Ending" if appropriate.

Add a Button and a Sequence of Event Handlers to Launch the Report

1. In the new form, select Edit>Component>New>Button.
2. Name the button GenerateReport, with caption s&Print.
3. On the Behavior tab, enter GenerateReport as the primary event.
4. Click the Primary button, and select New.
5. Use the drop-down button to select Type Run Background Task.
6. Click theParms button and then the Type Specific Parametersbutton.
7. For the Taskname, enter V(BGTaskName) and give this variable the value ReportDemo. (To get to the variable editor, put the cursor before the V in V(BGTaskName) and type Alt-V.)
8. For Task Parms, enter V(BGTaskParms) with the value V(OrderStarting), V(OrderEnding), V(OrderDateStarting), V(OrderDateEnding).

NOTE: For the report (or EXE or SP) to be able to run through the Background Queue, the task name must be in a variable called BGTaskName and the parameters must be in a variable called BGTaskParms.

If the parameter list is too long to fit into a single variable, break it into two or more variables and set the value of BGTaskParms to those variables. For example, set the value of BGTaskParms to (V(BGTaskParms1)V(BGTaskParms2), with BGTaskParms1 and BGTaskParms2 containing the actual report parameters. The actual report parameters may include the TaskMan keywords described on page 47.

9. Click OK until you return to the list of Event Handlers.
10. Click New to create another GenerateReport event, and select Run Script for its type.
11. Click theParms Button, and then the Type Specific Parametersbutton.
12. Click the script ReportSubmitted, without any parameters

Testing the Report

To check whether your report is working at this stage, try running it and checking the results:

1. Run the report, selecting some dates and clicking the Print button.
2. Open the Background Task History and Active Background Tasks forms to monitor the progress of your report.

Adding Date Offset Check Boxes

These steps add the Date Offset functionality to the form. The Customer Orders for Reservable Items Report form is an example of a SyteLine report that does this.

1. For each Date (or Date Range) add a check box:
2. For each Date field, create a hidden edit field:

   - Name: \textit{DateFieldOffset} - The name should match up to the date field name. If the name of the field is OrderDateStarting, the name of this field should be OrderDateStartingOffset.
   - Component Class: DateOffsetVar
   - Bind to variable - where Variable is \textit{DateFieldOffset} (use the same name as the component name).
   - Initial Value should be left blank (NULL)
   - Examples: Add OrderDateStartingOffset, bound to variable OrderDateStartingOffset. Add OrderDateEndingOffset, bound to variable OrderDateEndingOffset.

3. Add an event to the Print button to set the Offset fields:
   a. In the Behavior tab, create a Primary Event (there should already be an event for GenerateReport). Click the \textbf{Primary} button.
   b. In the Event Handlers window, click \textbf{New}. The event should already be filled in as GenerateReport.
   c. Set Type to \textbf{Run Script}.
   d. Click \textbf{Parms}.
   e. Click \textbf{Type Specific Parameters} and enter the following:
      - Name: SetDateOffset
      - Parameters: Date field name, Offset field name, Increment field name
      If there is more than one date field, enter all the date fields in this order: Date field, Offset field, Increment field. For example:
      OrderDateStarting,OrderDateStartingOffset,OrderDateIncrement,OrderDateEnding,OrderDateEndingOffset,OrderDateIncrement
   f. Click \textbf{OK} until you return to the Event Handlers window.
   g. In the Event Handlers window, move this up so it is the first event (use the up arrow button on the bottom right side of the screen). - this needs to run prior to run background task.

4. In the Task Parameter List (variable BGTaskParms), add the DueDateOffsets. For example, to the end of the list, you would add:

   V(OrderDateStartingOffset),V(OrderDateEndingOffset)
Adding Report Preview to the Form

If you are not using a standard SyteLine installation (for example, if you installed multiple pieces on a single machine), you will need to make sure TaskMan is set up so that report preview will work on your machine. See “Report Developer Installation” on page 43 for more information.

To add report preview functionality to a form:

1. Open the list of Event Handlers and set the following:
   - Event: PreviewReport
   - Description:
   - Sequence: 1
   - Type: Print Preview
2. Paste the following text into the Parms field:

   TASKNAME(V(BGTaskName))TASKPARMS(V(BGTaskParms))
   ERRORMESSAGE(mPrintPreviewError)

   Then click the Parms button and enter mPrintPreviewError as the error message.
3. Add a button to the form and enter PreviewReport as its primary event.
4. Keep clicking OK until you return to the form, and then save it.

Modifying the Form to Submit Tasks to the Background Queue

Set up the form to submit a task to the Background Queue:

1. Add a component called BackgroundQueue of type ObjMenuItem to the form. On the Behavior Tab, add RunBackgroundQueue as the primary event.
2. Create a RunBackgroundQueue event of type Run Form As Modal Child. The parameters for the event should be the following, where "GenerateReport" is the name of the Run Background Task event:

   BackgroundQueue( SETVARVALUES(BGTaskName=V(BGTaskName), RunTaskEvent=GenerateReport) )

3. Add the following parameters to the end of the parameter string for the Run Background Task event that submits the task. (For a report, usually this event is defined with the Print button.)

   TASKSTATUS(V(BGTaskStatus))TASKNUMBER(BGTaskNumber)

   TASKSTATUS is an optional input parameter to BGTaskSubmit. If the TASKSTATUS keyword is omitted, or if its value is anything other than WAITING, the task is inserted into the ActiveBGTasks table with status READY and will be run by TaskMan. If TASKSTATUS is set to WAITING, the task is entered in the ActiveBGTasks table with status WAITING. TaskMan ignores any records in this table with status other than READY or RUNNING.
TASKNUMBER is an optional keyword that specifies a variable name to hold the TaskNumber generated when a record is inserted into the ActiveBGTasks table. If this keyword is used, Infor ERP SyteLine will create the variable and set its value. These two optional parameters (taskstatus and tasknumber) are also available in the MGCore IDO and in the BGTaskSubmitSp stored procedure.

You can also include substitution keywords that are replaced by appropriate values after the task is submitted to TaskMan. For a list of the keywords, see “Infior Framework TaskMan Substitution Keywords” on page 47.

**Placing a Task on the Background Queue**

To add a report task to the background queue:

1. Open a form that can submit a background task to TaskMan (see the previous section).
2. Select **Actions>Background**.
3. Fill in the scheduling information on the Background Queue modal form. To test this feature, set the task for an hour from now.
4. To add the task to the background queue, click **OK**.

This sets the BGTaskStatus variable on the parent form to WAITING. TaskMan then generates the GenerateReport event, resets the BGTaskStatus variable, and retrieves the value of the BGTaskNumber variable from the parent form. The BGTaskNumber will be used in a call to SLJobQueues.BackGroundQueueSP in the Background Queue form.

**Checking the Status of the Report**

After you add a report task to the Background Queue, check the Active Background Tasks form to be sure the task is listed with a status of WAITING. After the time when the report is scheduled to run, check the Background Task History form to be sure the task is listed there. Verify its status, return codes, and parameters.

To check the tasks’ scheduling information, use SQL Enterprise Manager:

1. Open the SQL server containing the SyteLine database.
2. In the left pane, expand **Management>SQL Server Agent** and select **Jobs**.

   Your scheduled task should appear in the right pane, in the format **TaskName_TaskID**.
3. Right-click on the task and select **Properties**.
4. Select the Schedules tab and click the **Edit** button to display the scheduling information.
Using Infor Framework TaskMan

Infor Framework TaskMan, commonly called TaskMan, is a Windows service that:

- **Polls the application database** - TaskMan continuously polls all application databases to identify new tasks to run. The polling interval tells TaskMan how long to wait before polling the databases again. (Just leave the polling interval field blank if you want to accept the recommended default of 5000, which is 5 seconds). The application databases are polled in the order in which they were configured, and the polling interval that TaskMan uses is set by the first database TaskMan polls. To see the order in which the application databases were installed, run Registry Editor and navigate to HKEY_LOCAL_MACHINE\SOFTWARE\Mongoose\TaskMan. There should never be an application database that is monitored by two or more instances of TaskMan.

When a task is found that is waiting to be run, the task information (including executable, type of task parameters, and user name) is queried and the appropriate processing is performed.

- **Executes SQL stored procedures** - For a SQL stored procedure, TaskMan launches a database process thread. The database process then connects to the application database. It logs in as the SyteLine user who submitted the task. The stored procedure is then executed using the process connection.

- **Launches applications** - TaskMan launches a system process to execute any valid Windows application such as EDI (EDIImporter.exe) and APS (AIM_BATS.exe), enabling the application to carry out its processing. The application is executed under the user account used by the TaskMan service.

- **Generates Crystal Reports** - For a report, TaskMan launches RunReport.exe to execute a Crystal Report. The taskman.exe application then connects to the application database. It logs in as the SyteLine user who submitted the task. Once connected, it queries any user options for running the report, for example, output format or printer name. RunReport connects directly to the SQL database to query the strings table.

Reports are run just like all other executable applications. The RunReport.exe application handles the actual report generation. The RunReport.exe application uses the Report Design Component (RDC) to specify the database connection and output format for running the report. Once these options are set, it prints the report.

- **Runs IDO methods** - Instantiates a SyteLine IDO, and runs one of its methods.
Installation and Configuration

Standard Installation

TaskMan is installed and configured as part of the SyteLine installation. We recommend that you install TaskMan on the Utility Server. See the *SyteLine 7 ERP Installation Guide* for more information; however, much of the TaskMan setup is done behind the scenes. The standard installation process:

- Installs TaskMan on the server, which places the RunReport.exe program, the TaskMan.exe program, and other necessary files in the `installfolder\TaskMan` folder.
- Sets up the following folders within the TaskMan\Report folder:
  - **Errors.** Errors generated by Crystal Reports are written to this directory, under subfolders that match the IDs of the users who submitted the reports (for example, TaskMan\Errors\johsmi).
  - **Output Files.** The finished Crystal Reports output files are written to this folder, under subfolders that match the IDs of the users who submitted the reports. File extensions such as .DOC or .HTM indicate the format of the report.
  - **Reports.** All existing SyteLine report definitions (.rpt files) created in Crystal Reports are placed here during installation. Any additional report definitions you create should also be placed here.

Error and output file names include the site name and task number, in the format `taskname_site_tasknumber`. If TaskMan is configured to run against two or more databases with the same site specification (for example, test and production databases) and name clashes occur, a single digit is appended to the file name to eliminate the clash. For example: `APWirePostingReport_OH_150844_2`.

- Creates TaskMan as a Windows Service on the utility server, set for Automatic startup. If you enter a Service Logon and Password during the TaskMan Configuration part of the installation, that domain user ID and password are used to determine the Windows user account that controls the service. (You can also set the Service Logon and Password from the Services dialog after installation is complete. Go to **Start**>**Settings**>**Control Panel**>**Administrative Tools**>**Services.**)

- Sets up an ODBC data source name (DSN) connecting to the SyteLine application database. During the TaskMan Configuration part of the installation, you then configure TaskMan to connect to that DSN through ODBC, using SQL server authentication with the SQL database logins and passwords you provide.

The Service Configuration Manager utility also allows you to configure additional OLE DBs for monitoring by TaskMan. You can use this utility later by going to **Start**>**Programs**>**Infor**>**Setup**>**Server Service Configuration.** Online help is available.

- Determines which Strings table in the Forms database to use when displaying report form labels. This is a language-dependent setting that appears in the Sites/Entities form.
Report Developer Installation

Developers who are creating many custom reports may want to set up a separate, nonstandard test environment. If you do this, be sure the following requirements are met:

- After you install TaskMan, you must separately install Crystal Reports.
- On the Intranets form, the Task Man Path field should be set to the directory where TaskMan.exe and RunReport.exe are installed. If this is blank, or filled in incorrectly, the report will not preview. For additional information, see the description of the error “Missing RPT File” on page 59.
- On the Sites/Entities form, the Forms Database Name field should contain the name of the forms database. If this is filled in incorrectly, the report will preview but the labels will remain untranslated. For additional information, see the description of the error “Labels Not Replaced with Strings Table Values” on page 62.
- The General Parameters form should display the site ID. This site ID is used to select the appropriate Sites/Entities record. The Sites/Entities form should contain the Intranet name, which is used to select the Intranets record. If any of these links is incorrect, the report will not preview.
- The Crystal Reports DLLs must be on the same machine as MGCore.dll.

Setup the Utility Server to send E-mail Notifications

Configure the system by entering information into the following fields on the Intranets form:

- Server
- Server Port
- From Email
- Authentication
- Username
- Password
- Delivery Method
- Enable SSL
- Pickup Directory

Enable E-mail Attachments

1. Open the Intranets form within Infor ERP SyteLine.
2. Select your site.
3. On the Reports/TaskMan tab, ensure that the Send Email Notification field is selected.

Activate E-mail Notification

Perform the following tasks for each report that will send e-mail notifications:

1. Open the Report Options form within Infor ERP SyteLine.
2. For each report that will send e-mail notifications, select Yes in the E-mail Notification field.

NOTE: Depending on how you have organized your report options, you may have to create
Enter User E-mail Addresses
1. Open the Users form within Infor ERP SyteLine.
2. For each user who will receive e-mail report notification, ensure that a valid e-mail address has been entered.

Running Stored Procedures, Executables, and Methods Through Infor Framework TaskMan

In addition to running reports (described in the previous chapter), you can use TaskMan to run stored procedures, executables, or IDO methods.

Stored Procedures

TaskMan can execute stored procedures directly, without going through the IDO layer. To do this, TaskMan bundles the stored procedure in a transaction.

The following example shows the steps to set up a stored procedure that runs through Infor TaskMan from a form:

1. Set up a Background Task Definition record of type "SP" with the name of the stored procedure as the task name. For example, create a task called AddProcessErrorLogSp, of type SP.
2. Add a button to a form. The button will be used to run the stored procedure.
3. On the Behavior tab for the button, enter RunSp in the Primary Event field.
4. Click the Primary button.
5. Create a new event, selecting Run Background Task as the Event Type.
6. Click theParms button. Use mBackendMessage as the error message and sSubmitted as the success message.
7. Click the Type Specific Parameters button and enter AddProcessErrorLogSp as the Task Name. For TaskParms, enter BG~TASKID~,FV(TestMessages).
   The FV keyword tells SyteLine to enclose the value of TestMessage in single quotes.
The BG~TASKID~ substitution keyword is replaced with the task number. (See “Infior Framework TaskMan Substitution Keywords” on page 47 for a complete list of keywords.)
8. Save the event.
9. Add an edit box to the form with a variable called TestMessage as a data source.
10. Save the form.
11. Enter a message in the TestMessage edit box, and submit the task.
12. Open the Background Task History form. When the task completes, the test message should show up in the event log.
Executables

TaskMan can also be used to execute a command string as an operating system command shell. TaskMan takes the executable name from a Background Task Definitions record, appends the parameters from the SyteLine form, and attempts to execute the line.

The following example shows the steps to set up an executable that runs through TaskMan from a form:

1. Create a file called DeletePreviewFiles.cmd in your TaskMan folder. The text of this file should be as follows:
   
   ```
   FOR /D %%D IN ("taskman_dir\Report\OutputFiles\**") DO del /Q "%%D\Preview\**
   ```
   
   where `taskman_dir` is replaced by the name of your TaskMan folder.

2. Set up a Background Task Definition record:
   
   - Task Name: DeletePreviewFiles
   - Executable Name: DeletePreviewFiles.cmd
   - Executable Type: EXE

   If the executable is not in the TaskMan folder, include the path to the executable in the Executable Name field (for example, `c:\program files\MyExecutable.exe`).

3. Add a button to a form. The button will be used to run the executable.

4. On the Behavior tab for the button, enter **RunExe** in the Primary Event field.

5. Click the **Primary** button.

6. Create a new event, selecting **Run Background Task** as the Event Type.

7. Click the **Parms** button. Use **mBackendMessage** as the error message and **sSuccess** as the success message.

8. Click the **Type Specific Parameters** button and enter **DeletePreviewFiles** as the Task Name.

9. Save the event and the form.

10. When you click the button, this batch file will delete all the Print Preview intermediate files.

Database Connections

TaskMan will update ActiveBGTasks and BGTaskHistory for EXE tasks. It is up to the executable to handle all other database connections.

Returning Error Information

To get information back to TaskMan from an executable, use any of these methods in the EXE:

- Put a message in the ProcessErrorLog table. These messages appear in the Task Messages tab on the Background Task History form.
- Print the error message in a file called
  \textit{taskman-install-directory}\Output\task-name\task-number.txt
  
  (For example, c:\program files\Infor\Syteline\TaskMan\Output\APChecks_435.txt)
  TaskMan will use this as an error message in BGTaskHistory.

- Return an integer error code. TaskMan puts the EXE return code in the BGTaskHistory
table, with return code 0 indicating success.

\section*{IDO Methods}

The following example shows the steps to set up an IDO method that runs through
TaskMan from a form:

1. Set up a Background Task Definition record:
   - Task Name: Method name
   - Executable Name: Enter this in the format \textit{IDO.method} - for example,
     SL.SLExtfinParms.ExtFinExportAP
   - Executable Type: IDOMTH

2. In a form, set up an event whose task parameters match the IDO method’s parameters.
   You can pass bare values (for example, "MyParameter1,MyParameter2"). Note that
   Infor ERP SyteLine will not allow you to use \texttt{~LIT~} syntax as part of a value. If white
   spaces are significant, use the usual Infor ERP SyteLine keywords such as \texttt{P(...), V(...),}
   \texttt{C(...), and FPC(...), FV(...), or FC(...)}. 

3. Save the form and event.
Infior Framework TaskMan Substitution Keywords

TaskMan supports the following substitution keywords. Before executing a task, TaskMan replaces these keywords with their appropriate values when creating a string made up of the task executable name (defined in Background Task Definitions) plus the task parameters (passed from the SyteLine form).

Generally you specify these keywords when defining task parameters for an event on a form (see the note on page 37).

Some of these keywords are used to get values from TaskMan’s registry entries. These registry entries are in HKEY_LOCAL_MACHINE\SOFTWARE\Mongoose\TaskMan.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BG<del>TASKID</del></td>
<td>Replaced by the task number from the ActiveBGTasks and BGTaskHistory tables.</td>
</tr>
<tr>
<td>BG<del>TASKNAME</del></td>
<td>Replaced by the task name from the BGTaskDefinitions table.</td>
</tr>
<tr>
<td>BG<del>CONFIG</del></td>
<td>Replaced by the SyteLine configuration name.</td>
</tr>
<tr>
<td>BG<del>REQUSER</del></td>
<td>Replaced by the user name requesting the task.</td>
</tr>
<tr>
<td>BG<del>SQLLOGIN</del></td>
<td>Login associated with the user ID used to connect to a database.</td>
</tr>
<tr>
<td>BG<del>DSN</del></td>
<td>Name of the ODBC DSN used by TaskMan to connect to a database. This is taken from the Registry.</td>
</tr>
<tr>
<td>BG<del>UID</del></td>
<td>User ID for the ODBC used by TaskMan to connect to a database. This is taken from the Registry.</td>
</tr>
<tr>
<td>BG<del>SERVER</del></td>
<td>Server for the ODBC used by TaskMan to connect to a database.</td>
</tr>
<tr>
<td>BG<del>DB</del></td>
<td>Database for the ODBC used by TaskMan to connect to a database.</td>
</tr>
<tr>
<td>BG<del>DBNUM</del></td>
<td>Database's number, taken from the Registry.</td>
</tr>
<tr>
<td>BG<del>TMHOMEDIR</del></td>
<td>Directory where TaskMan.exe and RunCrystal.exe (or RunReport.exe) are installed.</td>
</tr>
<tr>
<td>BG<del>ERRFILE</del></td>
<td>When a task completes, TaskMan copies the contents of this file to BGTaskHistory, and then deletes the file.</td>
</tr>
<tr>
<td>BG<del>OUTDIR</del></td>
<td>Path to the output directory under the TaskMan home directory. This allows EXEs run through TaskMan to produce output.</td>
</tr>
<tr>
<td>BG<del>ISOLATIONLEVEL</del></td>
<td>The SQL Server isolation level used by TaskMan for reports and stored procedures. The return value is UNCOMMITTED or COMMITTED. Values are set in the Isolation Level field on the Background Task Definitions form and the Collection Read Mode field on the Process Defaults form.</td>
</tr>
<tr>
<td>BG<del>FAXNUM</del></td>
<td>The telephone number of the fax machine to which the document is sent. The keyword is replaced with the number specified in the Destination field on the Customer Document Profile form or the Vendor Document Profile form.</td>
</tr>
</tbody>
</table>
TaskMan also supports the following additional keywords used as command line switches.
TaskMan deletes these keywords from the task executable and parameter string.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
</table>
| BG~FAXSERVER~      | The name of the fax server machine. The keyword applies only to Windows Fax and Infor Framework Fax Service. If BG~FAXSERVER~ is omitted from a fax header or if the Fax Server field on the Intranets form is blank, then the default fax server name at run time is as follows:  
If Windows Fax is configured, then the server name is the name of the TaskMan machine. Windows Fax uses a modem on the TaskMan machine.  
If Infor Framework Fax Service is used, the server name is the name of the machine on which the service runs. |
| BG~FAXTOCOMPANY~   | The name of the company to which the fax is sent; the name is printed on the fax cover sheet. The keyword is replaced with the name specified in the Cover Sheet Company field on the Customer Document Profile form or the Vendor Document Profile form. The keyword does not apply to Windows Fax or Infor Framework Fax Service. |
| BG~FAXTONAME~      | The name of the individual to whom the fax is sent; the name is printed on the fax cover sheet. The keyword is replaced with the name specified in the Cover Sheet Contact field on the Customer Document Profile form or the Vendor Document Profile form. The keyword does not apply to Windows Fax or Infor Framework Fax Service. |
| BG~OUTPUTFILE~     | The report output file created in the directory <TaskMan_Directory>\Report\OutputFiles\<user> on the TaskMan machine. With Infor Framework Fax Service, the file is accessed directly from this directory. |

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BG<del>LEAVELITS</del></td>
<td>Infor ERP SyteLine wraps many values in <del>LIT</del>(...). This keyword, which can be specified in an event handler on a form, tells TaskMan to leave these values. For example, you may create an event on a form that runs an EXE through TaskMan. Use this keyword to allow any <del>LIT</del> keywords in the parameters to be passed through to the EXE.</td>
</tr>
<tr>
<td>BG<del>LEAVETEMPS</del></td>
<td>TaskMan normally puts the contents of the BG<del>ERRFILE</del> error file in BGTaskHistory and then deletes the file. This keyword tells TaskMan to leave the error files.</td>
</tr>
</tbody>
</table>
Running Infor Framework TaskMan in Debug Mode

If you are having problems with a background task, you can run TaskMan in debug mode. Doing so generates additional messages for the Microsoft Event Viewer.

Enable Debug Mode On Process Defaults Form

By using the Process Defaults form, you do not have to stop and restart the Infior Framework TaskMan service. For more information about the Process Defaults form, see the Infor ERP SyteLine online help.

1. Open the Process Defaults form.
2. In the Process Name field, select TaskMan Options.
3. In the Default Value field, enter `debug`.
4. Save the record and exit the form.

Enable Debug Mode By Restarting Service

1. On the server where TaskMan resides, open Windows Services.
2. In the list of services, select Infor Framework TaskMan.
3. If the Infor Framework TaskMan service is running, stop it.
5. In the Properties dialog’s Start parameters field, enter `debug`. Then click the Start button to restart TaskMan.
6. When you finish debugging the problem, be sure to stop the TaskMan service, remove the debug parameter, and restart it.

**CAUTION: When you stop TaskMan, all running tasks are terminated.**

TaskMan debug mode messages are listed and explained starting on page 73.

There is also an optional "nowait" parameter for TaskMan. If you are starting TaskMan manually, this keyword allows it to start faster.

RunReport debug mode messages are listed and explained starting on page 82.
Checking Infor Framework TaskMan Events in the Event Log

In addition to the debug messages mentioned above, TaskMan generates event messages during normal processing. You can view these messages in the Microsoft Event Viewer. Some of the more common messages are listed and described in “Event Messages from Infor Framework TaskMan” on page 70.

To access the Event Viewer, follow these steps on the server where TaskMan resides:

1. Select **Settings>Control Panel** from the Windows Start menu.
2. Open **Administrative Tools** and then **Event Viewer**.
3. Select **Application Log**.
Using SQL Profiler to Trace TaskMan Instances

NOTE: There can only be one instance of TaskMan monitoring an application database. Having more than one monitoring an application database can prevent report output from being placed in the output directory.

You can use SQL Profiler to see which instance of TaskMan is monitoring an application database.

1. Select Start>All Programs>Microsoft SQL Server 2005>SQL Server Management Studio.
2. Select Tools>SQL Server Profiler.
4. Connect to the application database server you want to trace.
5. On the Events tab, remove everything from the Scripted Event Classes column except TSQL -- SQL:Batch completed.
6. On the Data Columns tab, add Server Name to the Selected Data Column.
7. Click Run to start the trace.
### Specifying Report Options On the Process Defaults Form

You can use the Process Defaults form to enable several TaskMan options without having to restart the Infor Framework TaskMan service. For more information about the Process Defaults form, see the Infor ERP SyteLine online help.

<table>
<thead>
<tr>
<th>Process Default</th>
<th>Description</th>
</tr>
</thead>
</table>
| Report Output Obfuscation        | When this option is set to 1, report output is directed into this path: \TaskMan\Path\OutputFiles\ReportName_GUID.FileExtension  
All report files reside in the Output Files folder and are distinguished by the session ID appended to the report name.  
If this option is set to 0 (the default), report output is directed into the following paths, unless a Report Output Directory is specified in the Report Options form or as a process default (see that option below):  
\TaskMan\Path\OutputFiles\UserID\ReportName_Site_TaskNumber.FileExtension  
\TaskMan\Path\OutputFiles\UserID\Preview\ReportName_Site_TaskNumber.FileExtension (for preview) |
| Report Output Directory         | If a global file system/UNC path (for example \servername\ReportOutputs) is specified here, all report outputs are directed to this directory. However, if a Report Output Directory is also specified in the Report Options form, that specification overrides the process default.  
Within this directory, TaskMan looks for a user-specific subfolder; if not found, it creates the subfolder and directs report output into it. User-specific subfolders are used only if the Report Output Obfuscation process default (described above) is set to 0.  
You can specify the following TaskMan keywords within the output path, which are substituted accordingly:  
**BG~TMHOMEDIR~**  - TaskMan Home Directory  
**BG~TASKID~**  - Task ID/Number  
**BG~TASKNAME~**  - Task Name  
**BG~CONFIG~**  - Configuration Name  
**BG~REQUSER~**  - Task Requesting User |
### TaskMan Options

<table>
<thead>
<tr>
<th>Process Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>debug</td>
<td>Run the current TaskMan thread in debug mode. For more information, see “Infor Framework TaskMan Debug Mode Messages” on page 73.</td>
</tr>
<tr>
<td>debugrep</td>
<td>Creates a text file containing a log of messages generated during the running of a report.</td>
</tr>
<tr>
<td>eventlog</td>
<td>When debugging background tasks (with the debug option above), TaskMan directs its debug logs to the toolset messaging systems, which can be viewed through the IDO Runtime Host or Log Monitor utilities. If you want to direct TaskMan debug logs to the Windows event systems so you can view and debug them through the Windows Event Viewer, use this option.</td>
</tr>
<tr>
<td>taskmsg</td>
<td>If this is specified, TaskMan will insert some task-specific messages while processing requested tasks, allowing you to review the status/process of tasks. These messages can be viewed in the Task Messages area on the &lt;b&gt;Background Task History&lt;/b&gt; form.</td>
</tr>
<tr>
<td>UsernameFilter</td>
<td>user01, user02, etc.</td>
</tr>
<tr>
<td>TasknameFilter</td>
<td>taskname01, taskname02, etc.</td>
</tr>
</tbody>
</table>
This chapter describes how to troubleshoot problems with printing SyteLine reports, previewing reports, or using TaskMan. The information here is presented for developers, with solutions that may include fixing stored procedures or performing SQL queries. Some of the same troubleshooting information is also presented, without the developer-related sections, in the System Administration Guide.

The list of problems is organized according to general area. You might find it easier to locate a specific problem or error by using the Acrobat Reader "Find" feature with this online document.

Where to Find Error Message Information

You might find error messages or additional information about an error in the following places:

Background Task History Form
Error messages for reports and previews are placed in the Error Message field on the Background Task History form.

Error Log
If Crystal Reports captured an error message, see the error log for additional information. The error log is located in the TaskMan/Reports/Errors/userID subdirectory, under the SyteLine install directory on the server PC where TaskMan resides.

Event Viewer
TaskMan runs as a service under Windows and generates event messages that you can view in the Microsoft Event Viewer. Some common event messages are listed starting on page 70. If you are having problems with a background task, you can run TaskMan in debug mode, which generates additional messages. (See “Running Infor Framework TaskMan in Debug Mode” on page 49; “Infor Framework TaskMan Debug Mode Messages” on page 73; and “RunReport Debug Messages” on page 82.)
Crystal Reports Are Not Printing

If run-time users cannot print reports, consider the following possibilities:

- Make sure the Infor Framework TaskMan service is started on the utility server.
- Verify that TaskMan is set up as a Windows service that logs on as a User Account. SyteLine uses this account logon to access printers.
- Verify that the user account configured as the owner of the Infor Framework TaskMan service has print privileges under Windows for the default printer, plus all printers listed on the Report Options form.
- Verify that a default printer has been configured for the server on which TaskMan resides.
- If the report being printed requires a printer other than the default printer on TaskMan's server, verify that the correct printer is defined for the report on the Report Options form.
- Verify, through Server Service Configuration, that an OLE DB configuration is listed for the database to which the user is connected.
- The user account that was used to run the Infor Framework TaskMan service must be the same user account that was used to map the printers on the TaskMan machine.
- Verify that the printer was mapped using the printer name, not the share name.
- Use SQL Profiler to verify that only one instance of TaskMan is monitoring the application database. See “Using SQL Profiler to Trace TaskMan Instances” on page 51 for more information.
E-mail Notifications Are Not Being Received

If run-time users are not receiving e-mail messages about background tasks, consider the following:

- Verify that TaskMan is set up as a Windows service that logs on as a User Account. If TaskMan is configured as a Local System Account, e-mail notification cannot be sent.
- Verify that the User Account configured for the Infor Framework TaskMan service under Windows has e-mail privileges in your mail system.
- On the Intranets form, the Send Email Notification field must be selected. TaskMan gets the value that is set here for the first database TaskMan is configured to use, and then uses that value for all databases to which it connects from the utility server.

If you do not know which database is the "first" database, you can select the Send Email Notification field for another database; then no e-mail will be sent and no error messages appear in SyteLine or in the NT Event Logs. You must either determine what the "first" database is and set the flag there, or set the flag in all databases. Conversely, to stop e-mail notifications, either clear the flag in the "first" database or clear the flag in all databases.

- Verify that the recipient's user ID specifies an e-mail address in the Users form.
- The system requires a default e-mail profile, or a profile called "TaskMan," on the server where TaskMan resides. To verify this, if Microsoft Outlook is installed, open its properties (right-click on the Outlook icon and select Properties). Click Show Profiles. If there is no TaskMan profile, copy one that already exists, copy it, and rename it TaskMan.
- If the notification concerns a Crystal Report, verify that e-mail access is set up correctly for the report you are troubleshooting. For the procedure to verify that e-mail access is set up correctly, see .

Setting Up E-mail Access for Crystal Reports

If users and customers outside your network are not receiving automatically forwarded Crystal Reports, try the following possible solutions:

- If only certain users are not receiving the reports, but others are, verify that report options are set correctly for those users, by performing the following actions:
  a. On the Background Task Definitions form, click Report Options.
  b. For the report and user you are checking, verify that the E-mail Notification option is set to Yes.
     If Attach Report is set to Yes, you must also set E-mail Notification to Yes.
- If no users outside your network are getting the reports, verify that the Exchange Server through which the SyteLine system routes e-mail is set to relay e-mail, by performing the following actions:
  a. Open the Intranets form, Reports/TaskMan tab, and make note of the contents in the following fields:
     - SMTP Server
     - SMTP Server Port
SMTP From E-mail
b. Click Start > Run.
c. In the Open field, type command, and then click OK. The system displays a Windows Command Prompt window.
d. At the command prompt, enter the command: telnet
e. At the Microsoft Telnet command prompt, enter the correct version of the following command. If you are using a Windows XP or Windows Server 2003 system, enter: set localecho
f. At the prompt, enter the correct version of the following command, where SMTPserver is the name of the SMTP server you collected from the SMTP Server field and SMTPport is the port number you collected from the SMTP Port field in Step a:
   - If you are using a Windows XP system, enter: o SMTPserver SMTPport
   - If you are using a Windows Server 2003 system, enter: open SMTPserver SMTPport
g. At the SMTP server prompt, enter the command: hello
   The SMTP server responds with a message that ends with Hello, followed by the IP address.
h. Enter the command: mail from:<SMTPfrom>
   where SMTPfrom is the e-mail address you collected from the SMTP From E-mail field in Step a. The SMTP server responds with a message ending in ...Sender OK.
i. Enter the command rcpt to:<recipient_e-mail>
   where recipient_e-mail is the e-mail address of the user you are trying to relay the reports to.
   - If the SMTP Exchange server is not set up to relay e-mail, the server displays a message similar to the following:
     550 5.7.1 Unable to relay for recipient_e-mail
     If you see this type of message, configure your SMTP Exchange server to relay the e-mail. For the procedure, see your Exchange server documentation.
   - If the SMTP Exchange server is set up to relay e-mail, the server displays a message similar to the following:
     250 2.1.5 recipient_e-mail
     If you see this type of message, continue with the next step.
j. Enter the command: data
k. Type a test message to send to the recipient. When you are finished with your message, press the Enter key, followed by a period (.), followed by the Enter key again. This terminates the message and queues it for sending.
l. To exit the SMTP server session, enter quit, followed by any other key press.
m. To exit the Microsoft Telnet session, enter quit.
n. To exit the command prompt window, enter exit.
o. Confirm with the recipient that the test message was received.

NOTE: If you receive the error “Could not open connection to the host, on port [port #]” check the firewall and virus software on the mail server to see if they are blocking the e-mail.
Report Problems

There are several common problems that can occur when users try to preview or print a report.

Missing RPT File

If the "Missing RPT file" or "RPT file missing" error occurs for some, but not all, reports, check to make sure the specified RPT file is in TaskMan_Path\Report\Reports.

If this error occurs for all reports, first check to be sure you are not using an older version of Crystal Reports. If this is not the problem, there is probably a problem with the TaskMan Path value on the Intranets form. If you are using the Process Defaults form to temporarily redefine the Taskman path, make sure you also check the Report Output Directory field (on the Process Defaults form).

If you are attempting to preview a report over the internet, make sure the Report URL field, on the Intranets form, is populated, correct, and that you are able to connect to the URL.

Truncated Data in Text Output

When the report format is Text or Rich Text, if a field’s data does not fit within the field, Crystal Reports truncates the data. For left-justified text, the right side truncates. For right-justified text, the left side truncates. For center-justified text, both ends truncate. To prevent this, adjust the field size of the RPT file.

Changing the font size or selecting the Can Grow formatting option does not prevent data from being truncated.

File Not Found

File Not Found - Occurs for Certain (But Not All) Reports

If this error occurs for some, but not all, reports, it probably indicates an error in the programming of the report. Check for messages generated when the task ran in the Background Task History form, or check the Crystal Reports error log (TaskMan_Path\Report\Errors\user_directory). Either of these may give more information about the problem.

There are a number of things that can cause this error:

- The report's stored procedure was changed, but the RPT file wasn't resynchronized with the SP.

If the stored procedure's input parameters or output fields were changed, then open the Crystal RPT file and select Database>Verify Database. Sometimes a change to the body of the stored procedure will also cause this error. It is a good idea always to run Database>Verify Database after modifying a report stored procedure.
For Crystal Reports 9, it is a good idea to run the **Verify Database** twice in a row. The first time you run it, the system fixes the report; the second time you run it, you should get a "Report up to date" message. If you do not see this message, make sure the stored procedure does not have any of the problem features described in this section or in "RPT Files" on page 25.

- **A Transact-SQL error occurred in the SP.**

To get a more meaningful error message for a T-SQL error, run the report SP through SQL Query Analyzer. The easiest way to do this is to print the report and copy the report parameters from the **Background Task History** form. Paste these in as the parameters for the report SP in Query Analyzer, making these changes:

- Enclose character and date parameters in single quotes.
- Replace empty parameters with **NULL**.

If there is a T-SQL error, this will give a line number and error description. The most common SQL errors in report SPs are data truncation errors. Many report SPs use temp tables created using SQL datatypes. These tables are often populated from variables declared with user-defined datatypes. The lengths of many of these UDFs have been increased (for example, address fields for internationalization SiteId). This can cause a SQL exception by executing INSERT or UPDATE statements against the temp tables using these variables, or applying the CONVERT function to them.

- **The parameters passed by the SyteLine form are in the wrong order or are of the wrong datatype, which may produce a T-SQL error.**

For example, if a string or date parameter is plugged into an integer parameter, SQL generates an exception. Following the steps in the preceding bullet will catch these errors as well. If you run the report through Crystal, the order in which it prompts you for parameters must match the comma-delimited list of parameters passed from the SyteLine form.

- **You should also check the Background Task History form to make sure there are no uninterpreted V(…) or P(…) keywords in the parameters.**

Finally, in Crystal Reports, subreports are linked to the main report by parameters on the subreport. When Notes subreports are linked in, the ShowInternal and ShowExternal subreport parameters must be linked back to the appropriate Main report fields (typically, two main report parameters). If this is not done, Crystal will expect these values to be passed from the SyteLine form. If the wrong types of values are passed, you will get an error. If no values are passed, the report will probably hang when run through TaskMan, while Crystal tries to prompt for these values on the TaskMan machine.
The SP makes a call to RaiseErrorSp or raiserrorsp.

RaiseErrorSp and raiserrorsp are used to generate a SQL exception with a user-specified error message. Neither of these two stored procedures should be used in report SPs.

There are both ODBC-based reports and OLE DB (ADO)-based reports. With ODBC-based reports, using RaiseErrorSp or raiserrorsp produces an error like the following:

```plaintext
Error running report Error... [2003-10-01 15:39:31]
MainModule.Main:
Failed to open a rowset.
Details: ADO Error Code: 0x80040e14
Source: Microsoft OLE DB Provider for ODBC Drivers
Description: [Microsoft][ODBC SQL Server Driver][SQL Server]<MsgTag>Error Message<MsgTag>
SQL State: 37000
Native Error: 50000
```

With OLE DB (ADO)-based reports, using RaiseErrorSp or raiserrorsp produces an error like the following:

```plaintext
Error running report Error... [2003-10-01 15:53:51]
MainModule.Main:
Failed to open a rowset.
Details: ADO Error Code: 0x80004005
Source: Microsoft OLE DB Provider for ODBC Drivers
Description: [Microsoft][ODBC SQL Server Driver][SQL Server]Could not locate entry in sysdatabases for database 'HCL_App'. No entry found with that name. Make sure that the name is entered correctly.
SQL State: 08004
Native Error: 911
```

File Not Found - Occurs for All Reports

If this error occurs for all reports, the machine running reports probably has not been configured properly. Reports use MGCore.dll, which is installed on the Utility Server by default. However, in development environments, MGCore.dll may be running on users’ workstations.

To determine where MGCore.dll is running, you need to find the Infor ERP SyteLine Application ID and then determine whether the session manager server is used.

1. Run WinStudioAppIdEditor and select the application ID.
2. Edit that application ID. If the Session Manager Server field is blank, then MGCore.dll is running locally. Otherwise, this field contains the name of the utility server.

Check the following on the machine where MGCore.dll is run:
Crystal Reports must be installed. Installing Crystal Reports does not always configure all the export options. This may occur if you performed a "typical" install of Crystal Reports instead of a "complete" install. If you see the error message "missing or out-of-date export DLL," open any report and manually export it to the appropriate format to force Crystal Reports to finish configuring (the installation CD may be required for this).

This machine must have access to the TaskMan folder (where TaskMan.exe and RunReport.exe are installed) on the machine where TaskMan is running, or the TaskMan URL if previewing over the internet. The easiest way to check this is to select Run from the Windows Start menu and try to run a report, for example:

```
TaskMan Path\Report\Reports\ItemQuantitiesbyABCCode.rpt
```

In rare cases, this error occurs if the RunReport.exe has become corrupted. If this is the case, running the report through TaskMan produces an error code of -1,073,741,819 and the report fails.

Labels Not Replaced with Strings Table Values

**Finding the Strings Table Value through SyteLine**

To select the correct record on the Sites/Entities form, use the Site Name from the General Parameters form. Make sure that, for this site, there is a value in the Forms Database Name field of the Sites/Entities form (in some versions, labeled as the Strings Table Specification field). If the forms database is on a different server than the application database, the value for this field should also indicate the linked server name, in this format: server_name.Forms_database

TaskMan then determines the proper Strings table name by searching the specified Forms database for the Strings table associated with the current SyteLine session.

**Finding the Strings Table Value through a Query**

Developers can check this through SQL Server Query Analyzer. Run the following query against the application database:

```
SELECT i.intranet_name, p.site, i.tm_path , i.report_path, s.strings_table
FROM parms p
    INNER JOIN site s ON s.site = p.site
    INNER JOIN intranet i ON i.intranet_name = s.intranet_name
```

The strings_table field cannot be NULL. If this SELECT doesn't return any rows, then some piece of initialization data required by TaskMan was not entered.

**Validating the Strings Table Name through a Query**

To test the Strings table name through SQL Query Analyzer, run the following statement against the application database to return the number of strings in the Strings table (approximately 16000):
DECLARE
DECLARE @StringSQL AS NVARCHAR(255),
DECLARE @FormsDB AS NVARCHAR(255),
DECLARE @OwnerPos INT
SELECT @FormsDB = s.strings_table
FROM parms p INNER JOIN site s ON s.site = p.site
SELECT @OwnerPos = CHARINDEX(N'.dbo.', @FormsDb)
IF @OwnerPos <= 0 SELECT @OwnerPos = CHARINDEX(N'..', @FormsDb)
IF @OwnerPos > 0 SELECT @FormsDb = LEFT(@FormsDb, @OwnerPos - 1)
SELECT @StringSQL = N'SELECT COUNT(*)FROM ' + @FormsDB + N'.dbo.' +
  l.StringTableName
FROM  LanguageIDs l
WHERE l.LanguageID = 1033
EXEC (@StringSQL)

If the Strings table specification is invalid, this SQL code returns something similar to the following:

   Server: Msg 208, Level 16, State 1, Line 1
   Invalid object name 'String_Table_Specification'.

Invalid String Table Name

If you get an error message that says "Invalid String Table Name" followed by a number, check that you are running the proper version of Infor ERP SyteLine against TaskMan.

Formatting Problems When Exporting Report Output

When using an output format other than .RPT, certain features may not be supported by the Crystal Report export DLLs. For more information about unsupported export features, access the Crystal Reports Support Web page at www.businessobjects.com.
Other Infor Framework TaskMan Problems

TaskMan Service Does Not Start

If TaskMan does not start and you see this message in the Application Event log - "No database definitions defined. TaskMan must be configured before starting the service" - then consider the following:

- In Windows, verify your OLE DB (or ODBC) data source configurations for TaskMan.
- Run Service Configuration Manager and verify that the correct application databases are configured for TaskMan.
- TaskMan may be running under a user ID that does not have privileges to access the registry. Try restarting the service to run as a local system account. If it starts, then the problem is with the user ID.
- In Windows, restart Infor Framework TaskMan as an active service.

Another possibility if TaskMan fails to start: check the system log for an error from the Service Configuration Manager with the message "The Infor Framework TaskMan service failed to start due to the following error: The service did not start due to a logon failure."

TaskMan typically runs under a user ID. If the password for that user ID is incorrect, you will get this error message when trying to start TaskMan. To set the password, start Windows Services, select Infor Infor Framework TaskMan, and open up its properties. The password is on the Log On tab. If passwords are periodically reset, then you must reset this value when the passwords are changed.

Changes to Intranets Form Settings Are Ignored

When the Infor Framework TaskMan service starts, it caches information from the Intranets table. If you then change a setting on the Intranets form, TaskMan will not see this change. Instead, it continue to use the cached setting for the following fields:

- Polling Interval
- Connection Query Timeout
- Process Timeout
- Maximum Concurrent Tasks

For this reason, after changing one of these settings on the Intranets form, you must stop and restart the Infor Framework TaskMan service so the change will take effect.

No Output

Reports may be marked as Started and Completed in the Background Task History form, but there is no output in the ..\Program Files\Infor\SyteLine\Report\OutputFiles folder.

There are probably two or more instances of Infor Framework TaskMan monitoring the same database. To verify this, shut down the instance of Infor Framework TaskMan you think should be monitoring the database and resubmit the report. If the Background Task History record still gets updated, then there is at least one more instance of TaskMan monitoring the database. To find out if this is the case, launch the SQL profiler against the database to see what hosts were polling.
If you have this problem while previewing, it may be that the TaskMan path is incorrectly entered. If TaskMan is installed on the Utility server, the path on the Intranets form should be C:\<TaskManPath>, not \server\<TaskManPath>. If this is incorrect, the report will execute correctly, but PrintPreview.exe will not be able to monitor the folder and open the report output.

Intermittent Errors

The same report is run a number of times, sometimes running successfully, and other times erroring out. This happens for all reports. Whether it succeeds or fails appears to be random. This could occur because of the same problems described under the "No Output" heading. Alternatively, there may be two instances of TaskMan: one has the privileges needed to run reports, but the other does not.

Reports Fail with Return Code -1,073,741,819

If you see this return code, an executable may have become corrupt. Try double-clicking on RunReport.exe. If it does not open dialog boxes, the EXE have been corrupted, possibly by a virus. Delete it and copy it again from the installation CD.

Missing TMMsgs.dll

If TaskMan runs and processes background tasks, but all the event log messages start with the following message, then the TMMsgs.dll is missing:

The description for Event ID ( 100 ) in Source ( RSTaskMan ) could not be found. It contains the following insertion string(s)

TMMsgs.dll should be in the same folder as TaskMan.exe.

Batches of a Report Fail

If multiple requests for a particular report are submitted in batches, and the reports fail if more than 20 of them are submitted at once, then check the Max Concurrent field on the Background Task Definitions form. The value should be set to 20, since Crystal Reports does not allow more than 20 instances of a given report to be running at one time.

This problem typically occurs when users make a copy of one of the SyteLine reports and create a new Background Task Definition record for the new report, but do not reset the Max Concurrent field.

Notes Do Not Print on a Report

If you print a report, but the notes do not print along with it, they are probably not set up correctly. See “Adding Notes to Reports” on page 29.
No Default Set for Parameter $n$

You may see a warning message similar to the following:

```
WARNING: No default set for parameter $n$, userID <mailto:userID> setting parameter 38 =""; Error in File 
TaskMan_Path\Report\Reports\report_name.rpt>: operation illegal on linked parameter.
```

This error message occurs when the SyteLine form does not pass enough parameters to the Crystal RPT file. The most common cause is that a subreport with parameters (usually a notes subreport) has been linked in incorrectly. This is what caused the sample error message above.

The Notes subreports each have two parameter fields, ?@ShowInternal and ?@ShowExternal. These subreport parameter fields have to be linked back to the ?@ShowInternal and ?@ShowExternal parameters on the main report. Since the parameters were not linked back to the main report, Crystal Reports expected them to be supplied by the SyteLine form - but they were not.

When you get this error message, try to run the report through the Crystal Designer and have it prompt for parameters. The number and order of the parameters must be the same as in the SyteLine form. If it prompts for subreport parameters, you know there is some sort of link error.

Missing or Out-of-Date Export DLL

Installing Crystal Reports does not always configure everything needed to export reports. If you get the "Missing or Out-of-Date Export DLL" message, open any RPT file in Crystal Designer on the machine where MGCore.dll resides (usually the utility server, but it could be the local machine in some development environments). Select File>Export and then select the appropriate export format (for example, Crystal, HTML, etc.). Save the report to a disk file. This forces Crystal Reports to finish configuring the Export functionality. You do not need to do this for every report - just once in any report for each export format where you get this error. You may need to have the Crystal installation CD.

For more information about exporting reports, see your Crystal Reports documentation.

Background Task Runs But Has No History Record

A background task is submitted from a SyteLine form. It appears to have been submitted without any problems, but no record is created in the Background Task History form. The BGTaskHistory record is created by a trigger on the ActiveBGTasks table. ActiveBGTasks is the queue of tasks submitted to TaskMan. So, if there is no history record, the task never made it to the queue, despite any messages that may have displayed on the SyteLine form.

This is probably a bug in the sequence of form events the SyteLine form used to submit the task.
Error 13: Type Mismatch

There is a datatype mismatch between either what is passed from the SyteLine form to the RPT file, or from the RPT file to the stored procedure. This can occur if:

- A number field (for example, invoice number) is an NVARCHAR in the stored procedure, but a number in the RPT. If you perform a Set Datasource request, Crystal brings up a dialog where you can map stored procedure fields to report fields. Deselect Match by Type, and then map the number field in the stored procedure to the RPT. If the field is used in formulas, the report may still fail. You will have to correct the formulas (for example, String fields cannot be used in the ToText function).

- The SyteLine form passes the parameters in the wrong order, or a parameter is not defaulted correctly. For example, a form might be passing in a blank for the “Show Header” check box. If you get this error, try checking this check box and running it again. If this doesn’t work, try checking every check box and putting a value in every field. Also try running the stored procedure through SQL Query Analyzer with the same parameters as those in BGTaskHistory, replacing blank parameters with NULL, and putting single quotes around strings, dates, and guids.

Error 128: Error Running Report

If reports fail with this message, then:

- If the error message also includes the text “MainModule.Main: ” but no following text, this group section cannot be printed because its condition field is nonexistent or invalid. Format the section to choose another condition field.

One of the report groups is based on a field (or formula containing a field) that did not get mapped properly. Make sure all groups are based on fields that are in the stored procedure’s result set.

- If the error message also includes the text “Query Engine Error: ’42000:[Microsoft][ODBC SQL Server Driver][SQL Server]Invalid length parameter passed to the substring function,” this indicates that an ODBC-based report has to be converted to OLE DB (ADO). Reset the report’s datasource location (see “Resynchronizing a Report After Modifying a Stored Procedure” on page 27) and try running the report again.

- If the error message also includes the text “This field name is not known,” check for these possible problems:
  - An ODBC based report was not converted to OLE DB (ADO). Select Database>Set Datasource Location and try running the report again.
  - An RPT field was not successfully mapped to a stored procedure field (see the first bullet under “Error 13: Type Mismatch” for more information about this situation).

- If the error message also includes the text “The remaining text does not appear to be part of the formula,” there is a bug in an RPT formula. Try running the report through Crystal Designer with the same parameters to determine which formula is causing the problem.
Troubleshooting

Error 500: Not Enough Memory for Operation

There are a number of problems that can cause this error, including:

- A string formula longer than 254 characters
- Groups based on fields that do not exist. In the RPT file, select **Database > Verify Database**, and then try the report again.

If the problem persists, run the report through Crystal with the same parameters and see if it generates a more meaningful error message.

Error 515: Error Starting Report

You may see the following message:

```
Error in File ..\Infor\SyteLine\Report\OutputFiles\Report_Name.rpt: Error in formula Formula_Name. Formula_Text
```

The maximum length of a string that can be returned by a formula is 254 characters. Some of the formulas used for printing address fields can exceed this length, depending on the data. A bug in the formula may also cause this error.

Error 534: Error Detected by Database DLL

See the suggestions under “File Not Found” on page 59.

This Field Name Is Not Known

If running a report through TaskMan produces the error "This field name is not known," there are several things that may be wrong:

- This can happen if an ODBC based report was not converted to OLE DB (ADO). In Crystal Designer, set the datasource location (see “Resynchronizing a Report After Modifying a Stored Procedure” on page 27) and try running the report again.
- Having a RPT field which wasn't successfully mapped to an SP field, can also sometimes cause this. In these cases, the unmapped SP field will be part of the error message, for example, "This Field name is not Known : {Rpt_Ap01FRISp;1.TcAmtAmtPaid}"

Make sure the RPT field is in the stored procedure’s result set, and set the datasource location as described in the previous bullet.

Empty Columns Appear When Exporting to Excel

After outputting a report to Excel, there may be extra, blank columns. This is caused by excess space between columns in the RPT file. To modify the RTP and remove the excess space:

1. Note the position of the blank column.
2. Open the RPT file in Crystal Reports.
3. Remove some space between the column headers and/or details on either side of the blank column’s position.
4. Save the report and test the changes.
5. Continue removing space until the blank column no longer appears in the Excel output.

ActiveX Component Can’t Create Object

If you receive this error, it is most likely a problem with Crystal Reports. Reinstall Crystal Reports and apply all service packs.

The Transport Failed to connect to the server.

If taskman returns this error when you attempt to send an e-mail with the SMTP protocol, or if you receive a similar error when testing the SMTP protocol using Telnet (see “Setting Up E-mail Access for Crystal Reports” on page 57), the firewall or antivirus software on the mail server may be blocking the e-mail.
Event Messages from Infor Framework TaskMan

TaskMan runs as a service under Windows and generates event messages that you can view in the Microsoft Event Viewer. If you are having problems with a background task, you can run TaskMan in debug mode (see page 73), which generates additional messages for the Microsoft Event Viewer.

The following messages are generated normally and do not require TaskMan to be running in debug mode. If a database exception occurs, TaskMan tries to retrieve and log the error message.

<function name> failed with return code <code> in <TaskMan source file> at <Source file line number>.

Abnormal termination of Task <n> (returncode = <code>). Performing rollback.
A stored procedure background task was rolled back due to one of the following circumstances:
- The stored procedure generated a return code less than 0 or greater than 5.
- Executing the stored procedure generated an exception.

An unknown exception occurred. <TaskMan source file>: <Source file line number>

Canceling Process <taskname>, Task <n>
If TaskMan is shut down, an event log message will be printed as each running background task that is a stored procedure is canceled.

Canceling Report <taskname>, Task <n>
If TaskMan is shut down, an event log message will be printed as each running background task that is a report is cancelled. Some reports may be left hanging even if TaskMan is shut down.

Database <db>. RegCloseKey failed with return code <code>. <TaskMan source file>:< Source file line number>
RegCloseKeyEx failed when trying to open the HKEY_LOCAL_MACHINE Registry. TaskMan will shut down.

Database <db>. RegOpenKey failed with return code <n>. <TaskMan source file>:< Source file line number>
RegOpenKeyEx failed when trying to open the HKEY_LOCAL_MACHINE Registry. TaskMan will shut down.
Database <db>. RegQuery - DSN failed with return code <code>. <TaskMan source file>:< Source file line number>

RegQueryValueEx failed when trying to retrieve the data source name from the Registry. TaskMan will shut down. Use the TaskMan Configuration utility to check the configuration of data sources.

Database <db>. RegQuery - PWD failed with return code <code>. <TaskMan source file>:< Source file line number>

RegQueryValueEx failed when trying to retrieve the password from the Registry. TaskMan will shut down. Use the TaskMan Configuration utility to check the configuration of data sources.

Database <db>. RegQuery - UID failed with return code <code>. <TaskMan source file>:< Source file line number>

RegQueryValueEx failed when trying to retrieve the user ID from the Registry. TaskMan will shut down. Use the TaskMan Configuration utility to check the configuration of data sources.

DSN <dsn> Database <db>. Login successful

Informational message.

Either the DSN or the SQL login was not set. <ReturnMessage>

Informational message.

ODBC Time out expired. Max retries of <n> exceeded. Operation canceled

Informational message.

ODBC Time out expired. Retrying operation. Retry = <n>, Set Timeout = <n>

A timeout occurred while TaskMan tried to log on to a database or to maintain the background task tables in the application database.

RegisterServiceCtrlHandler failed with return code <n> in <TaskMan source file> at <Source file line number>.

RSTaskMan shutdown: <n> tasks are running and will be canceled.

If TaskMan is shutdown while tasks are running, these tasks will be cancelled.

RSTaskMan starting: Version <n>

Informational message.
RSTaskMan Terminating
Informational message.

RSTaskMan Terminating - TaskMan Home Directory not properly detected
TaskMan could not determine its home directory and will shut down.

SetServiceStatus failed in <TaskMan source file> at <Source file line number>.

StartServiceCtrlDispatcher failed with return code <n> in <TaskMan source file> at <Source file line number>.
TaskMan was not able to start the Service Dispatcher.

TaskTypeCode not recognized
The task type was not EXE, RPT, SP, or IDOMTH.
Infor Framework TaskMan Debug Mode Messages

The following messages are generated for the Microsoft Event Viewer only when TaskMan is running in debug mode. For more information, see “Running Infor Framework TaskMan in Debug Mode” on page 49.

**<SPname> After Call <TaskMan source file>:<Source file line number>**

This message is printed after a stored procedure is called.

**Active Task Set not open: <DatabaseInfo>. <TaskMan source file>:<Source file line number>**

TaskMan is trying to clear database connections

**Cannot find last slash. <TaskMan source file>:<Source file line number>**

Error while retrieving information about the home directory from which TaskMan is executing.

**Close process connection completed for Task <n>. <TaskMan source file>:<Source file line number>**

Informational message.

**Closing database: <dsn>.<TaskMan source file>:<Source file line number>**

Informational message.

**Closing process connection for Task <n>. <TaskMan source file>:<Source file line number>**

Informational message.

**Could not instantiate CIntranetSet. <TaskMan source file>:<Source file line number>**

TaskMan could not get the needed information from the Intranet table.

**Decrement <taskname> <tasknumber>. <TaskMan source file>:<Source file line number>**

Informational message - TaskMan is decrementing the Running Tasks list when the task completes.

**DELETE ActiveBGTasks where TaskNumber = <n>. <TaskMan source file>:<Source file line number>**

Informational message - the task is deleted from the active tasks table.
Deleting TaskInfo handle. <TaskMan source file>::<Source file line number>
Informational message - TaskMan is cleaning up report (RPT) process handles.

DSN <dsn> Database <db> Connecting DB<n>: DSN <dsn> User <userID>
Timeout: <n>. <TaskMan source file>::<Source file line number>
TaskMan is processing the list of configured databases and establishing connections to
them.

DSN <dsn> Database <db> DELETE ActiveBGTasks where TaskStatusCode = 'RUNNING'. <TaskMan source file>::<Source file line number>
When TaskMan first establishes a connection to a database, it deletes any background
tasks with a status of running.

DSN <dsn> Database <db>. Login successful. <TaskMan source file>::<Source file line number>
Informational message.

DSN <dsn> Database <db> Opening Active Tasks record set. <TaskMan source file>::<Source file line number>
Informational message.

DSN <dsn> Database <db> Num Databases <n>. Registry Information for
Database #<db> DSN <dsn> UID <userID> PWD **** String Table=<stringtable>
Poll=<n> Process Timeout=<n> Connection Timeout=<n> Max Num Tasks=<n>
Informational message about the database connection.

DSN <dsn> Database <db> Total number of Requests = <n>. <TaskMan source file>::<Source file line number>
Informational message.

Error retrieving TaskMan Module Name. <TaskMan source file>::<Source file line number>
TaskMan retrieves its module name in order to get its home directory. There was an error
in retrieving the module name, so TaskMan cannot determine its home directory.

Increment <taskname> <tasknumber> Total requests <n> Queue size <size>. <TaskMan source file>::<Source file line number>
TaskMan is polling the table for active background tasks in each configured application
database.
No Intranet records found, using defaults. <TaskMan source file>:<Source file line number>
TaskMan is using the default values because it could not find a matching Intranets record.

ODBC Time out expired. Max retries of <n> exceeded. Operation canceled.
<TaskMan source file>:<Source file line number>
The database process timed out, and TaskMan performed the allowed number of retries without success.

ODBC Time out expired. Retrying operation. Retry = <n>, Set Timeout = <n>.
<TaskMan source file>:<Source file line number>
The database process timed out. TaskMan will try again until it has exhausted the retry limit.

Opening Intranet record set. <TaskMan source file>:<Source file line number>
Informational message.

Pause to ensure SQL Server is completely up. <TaskMan source file>:<Source file line number>
TaskMan pauses to make sure that SQL Server has started before trying to access databases.

Poll=<n> Connect=<n> Process=<n> MaxNo=<n> NumRec=<n> site = <site>
Intranet = <intranetname> String Table = <stringtable> URL = <URLpath> Format = <outputformat> ReportPath = <path> Email Notif = <emailnotification>.
<TaskMan source file>:<Source file line number>
TaskMan successfully queried the Intranet table and retrieved the information listed here.

Rpt task failed. <TaskMan source file>:<Source file line number>
TaskMan failed while trying to run a task with an RPT extension.

ServiceMain starting. <TaskMan source file>:<Source file line number>
Informational message.

SQLCancel failed. Deleting TaskInfo handle. <TaskMan source file>:<Source file line number>
The SQL Cancel of the process and task was not successful.

SQLCancel of Task <tasknumber> completed. Return Code = <n>. <TaskMan source file>:<Source file line number>
TaskMan canceled a stored procedure background task.
Task <tasknumber> <taskname>. An error occurred while waiting for the process to finish. Error return = <code> message = <message>. <TaskMan source file>:<Source file line number>

A Windows error occurred.

Task <tasknumber> <taskname>. Call to AddProcessErrorLogSp failed for user <userID>. Return code = <code>. <TaskMan source file>:<Source file line number>

TaskMan tried and failed to add a task message to Background Task History.

Task <tasknumber> <taskname>. Call to CloseSessionSp failed for task <taskname> user <userID>. Return code = <code>, Error message = <message>. <TaskMan source file>:<Source file line number>

TaskMan could not retrieve the Report Options information.

Task <tasknumber> <taskname>. Call to GetTaskOptionsSp failed for task <taskname> user <userID>. Return code = <code>, Error message = <message>. <TaskMan source file>:<Source file line number>

Task <tasknumber> <taskname>. Call to InitSessionContextSp failed for task <taskname> user <userId>. Return code = <code>, Error message = <message>. <TaskMan source file>:<Source file line number>


Informational message.

Task <tasknumber> <taskname>. Calling sp.Call <TaskMan source file>:<Source file line number>

Informational message.

Task <tasknumber> <taskname>. Can transact. <TaskMan source file>:<Source file line number>

TaskMan could not start a stored procedure background task.

Task <tasknumber> <taskname>. Cannot transact. <TaskMan source file>:<Source file line number>

TaskMan cannot start a transaction for a stored procedure task. The stored procedure is not executed.
Task <tasknumber> <taskname>. CloseSessionSp called with Input Parameter <sessionID>. <TaskMan source file>:<Source file line number>
Informational message

Task <tasknumber> <taskname>. Could not close session <sessionID>. <TaskMan source file>:<Source file line number>

Task <tasknumber> <taskname>. Creating directory <directory>. Result = <code>. <TaskMan source file>:<Source file line number>
Informational message


Task <tasknumber> <taskname>. DELETE ActiveBGTasks where TaskNumber = <n>. <TaskMan source file>:<Source file line number>
Informational message - the task is deleted from the active tasks table.

Task <tasknumber> <taskname>. Error <code> <message> when starting Command - <commandline>. <TaskMan source file>:<Source file line number>
An error occurred when TaskMan was starting the specified command.

Task <tasknumber> <taskname>. Error moving file <outputfile> to <new outputfile>. <TaskMan source file>:<Source file line number>
TaskMan could not copy the output file to the OutputFile folder. Be sure permissions are set up properly, and the folder exists.

Task <tasknumber> <taskname>. Exiting RunTask: Removing Task from List. <TaskMan source file>:<Source file line number>
The background task has finished running.

Task <tasknumber> <taskname>. Fax=<faxname>, Fax Server = <faxserver>, Output Format=<outputformat>. <TaskMan source file>:<Source file line number>
Informational message.

Task <tasknumber> <taskname>. GetExitCodeProcess return = <n>. <TaskMan source file>:<Source file line number>
This message displays the return code for an executable program background task.
Task <tasknumber> <taskname>. GetTaskOptionsSp called with Input Parameters <taskname>, <userID>, <stringID> returned Output Parameters format = <output format>, printer = <printer name>, email = <email notification>, attach = <attach report>, email address = <email address>, Return code = <code>, Error message = <message>, String Table = <string table>, Fax Server = <fax server>.<TaskMan source file>:<Source file line number>

TaskMan is retrieving specific information about the run-time user requesting this report background task. This information is entered in the Report Options and Intranets forms.

Task <tasknumber> <taskname>. InitSessionContextSp called with Input Parameter <taskname>, returned Output Parameter <sessionID>, Return code = <code>. <TaskMan source file>:<Source file line number>

Informational message.

Task <tasknumber> <taskname>. InitSessionContextSp failed. Could not call SP. <TaskMan source file>:<Source file line number>

Task <tasknumber> <taskname>. MAPI Init failed. <TaskMan source file>:<Source file line number>

E-mail services could not be started.

Task <tasknumber> <taskname>. No attachment sent: either the report was sent to the printer, the Task Type was not RPT, or the report didn’t complete successfully. <TaskMan source file>:<Source file line number>

Task <tasknumber> <taskname>. Opening user names failed. <TaskMan source file>:<Source file line number>

TaskMan encountered a problem attempting to open the UserNames database table.

Task <tasknumber> <taskname>. Password decrypted. <TaskMan source file>:<Source file line number>

Informational message.

Task <tasknumber> <taskname>. Profile Values -- String Table: <stringtable>, String ID: <stringID>, Fax: <fax name>, Email: <email address>, Number of copies: <n>, Printer: <printer name>. <TaskMan source file>:<Source file line number>

Informational message.

Task <tasknumber> <taskname>. Return = <code> WaitForSingleObject return = <code>. <TaskMan source file>:<Source file line number>

This message is entered immediately after a background task has ended.
Task <tasknumber> <taskname>. Running: <stored procedure>.<TaskMan source file>:<Source file line number>

Informational message.

Task <tasknumber> <taskname>. Running EXE: <command line>. <TaskMan source file>:<Source file line number>

Informational message.


This is the command line TaskMan executes to run a Crystal report. For debugging purposes, you can paste the following command line into a batch file and execute it:

RunReport.exe <Command line>

Task <tasknumber> <taskname>. Sending Fax: <command line>

Informational message.

Task <tasknumber> <taskname>. Setting Connection Timeout <n> and opening a connection for this task.<TaskMan source file>:<Source file line number>

TaskMan opened a connection for this background task that will be used to update the Task History table and to delete entries from the Active Task table in the application database.

Task <tasknumber> <taskname>. Setting Process Timeout <n>. <TaskMan source file>:<Source file line number>

Informational message.

Task <tasknumber> <taskname>. Stored Procedure call GetSQLServerLoginSp failed with return code <code> and error message <message>. <TaskMan source file>:<Source file line number>

TaskMan could not get the SQL login information needed to process the task.

Task <tasknumber> <taskname>. Successful termination of Task. Performing commit. <TaskMan source file>:<Source file line number>

A stored procedure background task can be committed.

Task <tasknumber> <taskname>. Unable to access <document path>. Fax not sent. <TaskMan source file>:<Source file line number>

TaskMan could not access the RTF file using the specified path. Check to see that the file and path exist and that permissions allow TaskMan access to it.
Task <tasknumber> <taskname>. Unable to access <SendFax.exe path>. Fax not sent. <TaskMan source file>:<Source file line number>

TaskMan could not access the SendFax.exe utility using the specified path. Check to see that the file and path exist and that permissions allow TaskMan access to it.

Task <tasknumber> <taskname>. UPDATE ActiveBGTasks SET TaskStatusCode = 'RUNNING' where TaskNumber = <n> .<TaskMan source file>:<Source file line number>

As TaskMan selects a background task to run, it changes its status to running.

Task <tasknumber> <taskname>. UPDATE BGTaskHistory SET CompletionDate = <date>, CompletionStatus = <status>, TaskErrorMsg = '<message>' where TaskNumber = <n>. <TaskMan source file>:<Source file line number>

Informational message.

Task <tasknumber> <taskname>. UPDATE BGTaskHistory SET ProcessId = <pid> WHERE TaskNumber = <n>. <TaskMan source file>:<Source file line number>

This message displays the process ID (pid) for a background task. You can use the process ID to trace the status of a background task in the Windows Task Manager. This ID is also displayed in the Background Task History form.

Task <tasknumber> <taskname>. UserNames.Username = '<userID>'.<TaskMan source file>:<Source file line number>

The user ID of the run-time user requesting the current background task.

Task <tasknumber> <taskname>. Usernames.Username = <userID> Groupname = <groupname>.<TaskMan source file>:<Source file line number>

The user ID and group name of the run-time user requesting the current background task.

TaskMan Error: TaskCounter is not keeping accurate count of Running Tasks. <TaskMan source file>:<Source file line number>

The number of running tasks in the system does not match the task counter value.

TaskMan home directory: <homedirectory>. <TaskMan source file>:<Source file line number>

Informational message.

TaskMan Stopping: Clearing database connections. <TaskMan source file>:<Source file line number>

Informational message.
The maximum number of concurrently running tasks is <n>. <TaskMan source file>:<Source file line number>
Informational message.

<dsn> UID <userID> PWD **** String Table=<stringtable> Poll=<n> Process Timeout=<n> Connection Timeout=<n> Max Num Tasks=<n>. <TaskMan source file>:<Source file line number>
Informational message.

UPDATE BGTaskHistory SET CompletionDate = <date>, CompletionStatus = <status>, TaskErrorMsg = '<message>' where TaskNumber = <n>. <TaskMan source file>:<Source file line number>
Informational message.
RunReport Debug Messages

When Infor Framework TaskMan is running with the "debugrep" startup parameter, it creates a text file containing a log of messages generated during the running of a report. For more information, see the note about debugrep under "Running Infor Framework TaskMan in Debug Mode" on page 49.

Sample Debug File

Following is a sample debug file, with notes explaining the information in each section. The notes will be referred to later, in "Using the RunReport Debug Files."

A. This indicates the report's start date and time.

** 4/23/2009 10:17:34 AM**

COMMAND LINE PARAMETERS:
- -d "SL8.00_OH_Dev" -l syteline -s SyteLineDev_Forms.dbo.Strings -f 7 -r ItemQuantitiesbyABCCode -t 923861 -u wilala -e "D:\TaskMan\Report\Errors\wilala\ItemQuantitiesbyABCCodeReport_OH_923861.txt" -site "OH" -db OH_App -svr CATBIRD -ptr \uscmhprn1\custhp -debug -m ~LIT~(A)~LIT~(B)~LIT~(C),~LIT~(M)~LIT~(P)~LIT~(T),~LIT~(M)~LIT~(T)~LIT~(F)~LIT~(O),~LIT~(1),~LIT~(1),~LIT~(),~LIT~(),~LIT~(),~LIT~(A),~LIT~(B),~LIT~()

These are the parameters TaskMan.exe formats and passes to RunReport.exe.

A. This indicates the report's start date and time.

** 4/23/2009 10:17:34 AM**

A. This indicates the report's start date and time.

B. This is a list of some of the parameters RunReport.exe has parsed from the command line.

** SETTING REPORT LOGONS AND STRINGS **

C. RunReport sets the main report logon information, and then loops through the report sections, extracting string names from formulas, and setting subreport logon information.

Setting Logon for : Rpt_ItemQuantitiesbyABCCodeSp;1 Original table location: Rpt_ItemQuantitiesbyABCCodeSp;1
Old CRAXDRT.ConnectionProperties values
Provider=SQLOLEDB  
Data Source=CATBIRD  
Initial Catalog=OH_App  
User ID=sa  
Integrated Security=False  
Locale Identifier=1033  
Connect Timeout=15  
General Timeout=0  
OLE DB Services=-5  
Current Language=  
Initial File Name=  
Use Encryption for Data=0  
Replication server name connect option=  
Tag with column collation when possible=0

D. For each Stored Procedure (or table or view) used as a data source in the main report and in subreports, RunReport loops through all the original connection properties that were saved with the original report. This information about the original connection information usually is not very useful; it was inserted for specific problems that only show up if the name of the database where the report was written is different from the name of the database where it is run. The preceding text allows you to make sure that is the case.

Using OLE DB (ADO) to connect directly to a server: CATBIRD

E. The toolset supports OLE DB (ADO) connecting directly to SQL SERVER, or connecting to an ODBC data source. The toolset also supports using an ODBC data source. The Provider connection property below specifies which of these is used. The following lines give a more readable version of the Provider and Data Source fields.

New Location: Rpt_ItemQuantitiesbyABCCodeSp;1  
Set from value: OH_App.dbo.Rpt_ItemQuantitiesbyABCCodeSp;1

New CRAXDRT.ConnectionProperties values
Provider=SQLOLEDB  
Data Source=CATBIRD  
Initial Catalog=OH_App  
User ID=syteline  
Integrated Security=False  
Locale Identifier=1033  
Connect Timeout=15  
General Timeout=0  
OLE DB Services=-5  
Current Language=  
Initial File Name=  
Use Encryption for Data=0  
Replication server name connect option=  
Tag with column collation when possible=0

F. The code that sets the logons also resets the database name.

New connection OK

G. In debug mode, RunReport tests the database logon information after resetting it.
String Table Query:

```
SELECT 'BG~USERID~' AS Name, 'wilala' AS String, 1 As ScopeType, 0 As OrderFlag UNION SELECT 'BG~UID~' As Name, 'wilala' AS String, 1 As ScopeType, 0 As OrderFlag UNION SELECT 'BG~SITEID~' As Name, ISNULL([site], N'') AS String, 1 As ScopeType, 0 AS OrderFlag FROM parms UNION SELECT 'BG~COMPANYNAME~' AS Name, ISNULL([Company], N'') AS String, 1 AS ScopeType, 0 AS OrderFlag FROM parms UNION SELECT Name, ISNULL(String, N''), ISNULL(ScopeType, 4), 0 AS OrderFlag FROM SyteLineDev_Forms.dbo.Strings WHERE Name IN ('sWarehouse','sABC','sItem','sDescription','sType','sSource','sStock','sLeadTime','sLotSize','sSafetyStock','sU/M','sOnHand','sOnOrder','sWIP','sListYesNo=1','sListYesNo=0','sDays','sAllocOrder','sAllocJob','sAllocProd','fItemQuantitiesbyABCCodeReport','sMaterialType','sStocked','sNotStocked','sPrintZeroQtyOnHandItems','sProductCode','sStarting','sEnding','sABCCode','sSource','sItem','sYes','sNo','sYes','sNo','sYes','sNo','sPage','sOf') AND Name IS NOT NULL AND ScopeType < 3 ORDER BY Name, OrderFlag ASC, ScopeType DESC
```

H. After looping through all of the formulas to extract the string names, RunReport constructs a SELECT to retrieve them from the Forms database. If strings are not getting translated, this select can be run in query analyzer against the Application database to see if it returns any strings.
SETTING REPORT PARAMETERS
Parameter# 1 does not need value : {?@ABCCode}, value passed from WinStudio = ABC
Parameter# 2 does not need value : {?@Source}, value passed from WinStudio = MPT
Parameter# 3 does not need value : {?@MaterialType}, value passed from WinStudio = MTFO
Parameter# 4 does not need value : {?@Stocked}, value passed from WinStudio = 1
Parameter# 5 does not need value : {?@NotStocked}, value passed from WinStudio = 1
Parameter# 6 needs value, set to null : {?@ZeroQty}, value passed from WinStudio =
Parameter# 7 needs value, set to null : {?@WarehouseStarting}, value passed from WinStudio =
Parameter# 8 needs value, set to null : {?@WarehouseEnding}, value passed from WinStudio =
Parameter# 9 does not need value : {?@ItemStarting}, value passed from WinStudio = A
Parameter# 10 does not need value : {?@ItemEnding}, value passed from WinStudio = B
Parameter# 11 needs value, set to null : {?@ProductCodeStarting}, value passed from WinStudio =
Parameter# 12 needs value, set to null : {?@ProductCodeEnding}, value passed from WinStudio =
Parameter# 13 does not need value : {?@DisplayHeader}, value passed from WinStudio = 1

I. RunReport takes the report parameter string, indicated by the "-m" command line switch, and pulls the parameters out one at a time and tries to plug them into the report's parameters. The parameter string is a comma-delimited list of parameters. If a comma occurs within a ~LIT~(… , …) it is not treated as a parameter delimiter; it is treated as part of the parameter. RunReport works its way through the parameter string, pulling out a parameter, processing the ~LIT~ keywords, and then setting the value of the corresponding report parameter.

If an empty string is passed as a parameter, then the report parameter is set to NULL. If there are more report parameters than there are parameter values passed from the WinStudio form, then the remaining Report parameters are set to NULL.

EXPORTING REPORT. SQL Query String:
"OH_App"."dbo"."Rpt_ItemQuantitiesbyABCCodeSp";1 N'ABC', N'MPT', N'MTFO',
1, I, NULL, NULL, NULL, N'A', N'B', NULL, NULL, 1

J. After setting the parameters, Crystal formulates a SQL statement to retrieve the report's data. This statement can be executed in Query Analyzer.

PRINTER PROPERTIES:
Printer = \uscmhprn1\custhp
Driver  = HP LaserJet 5N
Port    = custhp

K. RunReport calls the Windows API to retrieve the printer driver and port from the printer name. If this fails, either there is an accessibility issue which will prevent TaskMan from sending output to this printer or there is a typo in the printer name as entered in ReportOptions.
NEW REPORT PRINTER PROPERTIES:
  Printer = \uscmhprn1\custhp
  Driver  = winspool
  Port    = Ne01:
L. RunReport makes a Crystal RDC method call to set the printer name, driver, and port. If these values are blank, the report will go to the default printer on the TaskMan machine.

REPORT FINISHED
M. If the statement "REPORT FINISHED" isn't at the end of the debug file, the task may be hanging, or there may have been an error that RunReport could not catch.

Using the RunReport Debug Files
The following procedures refer to the notes in the preceding sample debug file.

Strings Are Not Being Translated
Check the information above notes B and C to make sure the string table got passed in properly. The string table name can be checked by logging into the Application database through SQL Query Analyzer and running the following SELECT statement, replacing string_table with the value passed to RunReport by the -s command line switch.

```
SELECT TOP 10 * FROM string_table
```

This should return 10 rows. It typically fails if there is a problem with the setup. It will also fail if a translated string table is missing.

The SELECT statement RunReport used to retrieve strings is shown above note I. This statement also can be run in Query Analyzer to see if it returns the appropriate strings.

Incorrect Report Data
A common problem for reports is that parameters are entered in the report form, but the report comes back either empty or displaying every possible record. The parameters passed from the SyteLine report form are displayed on the Background Task History form. SyteLine automatically processes the ~LIT~ keywords before displaying the parameter string shown on the history form. To see these keywords, run a query against the BGTaskHistory table using SQL Query Analyzer. However, usually just checking the parameters in the history form will do the trick.

Next, check the information above notes B and C to see if the parameter string was passed to RunReport and parsed properly. The section labeled SETTING REPORT PARAMETERS (note J) shows whether the ~LIT~ keywords were processed correctly and the report parameters are either set to these values or set to NULL.

Finally, the information above note K should show all of the parameters embedded in a SQL statement. Executing this statement will show whether the problem is in the RPT file or the stored procedure.
Output to the Default Printer Instead of the Report Options Printer

If reports are not printing, check the information above notes B and C to make sure the printer name was passed correctly. The PrinterName above note C should be the value specified in the Report Options form.

Log into the TaskMan machine using the TaskMan ID; then copy the PrinterName value into **Start>Run**. If doing this opens the printer queue, then there are no typos in the printer name and TaskMan can access the printer.

RunReport uses the printer name to get the printer driver and port using Windows API calls. The information above note I should show the results from making these calls. Typically, Windows throws an error if these API calls fail. RunReport will catch the error and put it in the Background Task History form.

RunReport uses the printer name, driver, and port in the Crystal SelectPrinter RDC method call. If this call succeeds, then the report's new printer properties should appear above note M. If the call fails, the Crystal RDC method does not generate an error; however, the printer name, driver, and port will be blank in the section above M. This causes the output to be sent to the default printer instead of the printer specified in the Report Options form.

Incorrect Database Name

The debug information above note G is printed after RunReport tries to reset the database name. The Crystal API does not generate an error if this fails. If the wrong database name appears in an error message in the Background Task History form, check the "Set from value" above note G to make sure RunReport was using the appropriate value for the database name.

For OLE DB(ADO) reports, the database name is the "Initial Catalog" value of the connection properties. Make sure the original Initial Catalog (above note E) has been updated in the New Connection Properties section (above note G). Finally, check the information above note K to make sure the correct database name was used in the SQL query Crystal used to retrieve a record set.
Using RunReport

Infior Framework TaskMan launches a system process (RunReport.exe) to execute a Crystal Report. The RunReport application then connects to the application database. RunReport connects directly to a SQL server using OLE DB (ADO) and logs in using the SQL login associated with the SyteLine user who submitted the task.

Once connected, RunReport queries any user options for running the report, for example, output format or printer name. RunReport then uses the Crystal Report Designer Component (RDC) to specify the database connection, output format, and other options for running the report. Once these options are set, RunReport prints the report.

Generally, RunReport is called from the SyteLine application; however, you can also run RunReport from the command line, specifying connection options, the report name, and report options. This could be used, for example, to run reports directly from an add-on product, or to run reports immediately from a workflow on a client (without having to put the report on a background queue) and then do something with the report's output in the workflow.

However, if you execute RunReport from a command line, you cannot take advantage of TaskMan’s task history and management features.
MGReportProcessor

Core report processing is handled by MGReportProcessor, for which RunReport is a wrapper to call methods in MGReportProcessor. This allows you to create your own executable, being processed by TaskMan, which could access report execution functionality in MGReportProcessor.

RunReport from the Command Line

Syntax

The command line syntax is this:

RunReport.exe switches

where switches must be preceded by either minus ( - ) or forward slash ( / ). If the -m switch is used, it must be the last switch; the order of the other switches does not matter. Any switch that is used must be followed by the appropriate value. All switches, other than -m, require their values to be enclosed in parentheses if the value contains blank spaces. The value following -m must not be enclosed in parentheses.

Switches

The following switches may be used in the RunReport.exe command line:

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h (optional)</td>
<td>TaskMan home directory, where taskman.exe runs. If this parameter is omitted, RunReport uses its own directory.</td>
</tr>
<tr>
<td>-t (optional)</td>
<td>Task number. For reports run through TaskMan, this is the Task Num field on the Active Background Tasks form, and the Task Number field on the Background Task History form. This number is generated by SQL when a record is inserted into the ActiveBGTasks table.</td>
</tr>
<tr>
<td>-r</td>
<td>Report name. If the report name does not include a path, RunReport tacks on HomeDirectory\Report\Reports\ to the report name.</td>
</tr>
<tr>
<td>-d (optional)</td>
<td>DSN for the ODBC that RunReport uses to connect to the SyteLine database. This is the same as the TaskMan ODBC. Use this parameter plus the -db parameter if you are using either ODBC (RDO) or OLE DB (ADO) to connect to an ODBC data source to access the database. This -d parameter is ignored if you are using a direct OLE DB (ADO) connection to the SQL database.</td>
</tr>
<tr>
<td>-p (optional)</td>
<td>Password for the ODBC connection - or for the SQL database, if you are using a direct OLE DB (ADO) connection. This value defaults to the empty string.</td>
</tr>
<tr>
<td>-l (optional)</td>
<td>Login for the SQL database, if you are using a direct OLE DB (ADO) connection. This value defaults to sa.</td>
</tr>
<tr>
<td>-u (optional)</td>
<td>SyteLine user ID for the person submitting the report. This is the User ID field on the Users form. It is also displayed on the Background Task History form, for tasks submitted to TaskMan. This value defaults to the ODBC login.</td>
</tr>
<tr>
<td>Switch</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>-db</td>
<td>Name of the application database. This is used to set the location for the main report and subreport data sources. If omitted, the data sources are left as is.</td>
</tr>
<tr>
<td>-g</td>
<td>User’s group. If provided, this is used in a SQL WHERE clause when querying the Strings table. It is used to filter on the ScopeName (the user’s group) and the ScopeType (2, for Group).</td>
</tr>
<tr>
<td>-session</td>
<td>Session identifier, passed when TaskMan calls RunReport, to indicate the session whose variables may be accessed by the report’s stored procedure.</td>
</tr>
<tr>
<td>-s</td>
<td>Name of the strings table used to translate report labels. RunReport uses an ODBC connection to the application database to query the Strings table, so the table name must be qualified enough for the query to work.</td>
</tr>
<tr>
<td></td>
<td>If the Strings table is in a forms database on the same server as the application database, then the strings table name must include that database name (for example, <code>Formsdb.dbo.StringTable</code>).</td>
</tr>
<tr>
<td></td>
<td>If the Strings table is in a database on a different server, then this server must be a linked server, and the Strings table name must include both the server name and the forms database name (for example, <code>Server.Formsdb.dbo.StringTable</code>).</td>
</tr>
<tr>
<td></td>
<td>If this parameter is omitted, the report will run, but the labels will not be translated and will be the actual string name, and a warning message will be placed in the error log.</td>
</tr>
<tr>
<td>-f</td>
<td>Integer code for the report output format:</td>
</tr>
<tr>
<td></td>
<td>1  Crystal Report (RPT) - the default value</td>
</tr>
<tr>
<td></td>
<td>2  Acrobat Format (PDF)</td>
</tr>
<tr>
<td></td>
<td>3  Comma Separated Values (CSV)</td>
</tr>
<tr>
<td></td>
<td>4  Excel 8.0 (format used with Excel 97 and Excel 2000)</td>
</tr>
<tr>
<td></td>
<td>5  Excel 8.0 extended (format used with Excel XP and Excel 2003)</td>
</tr>
<tr>
<td></td>
<td>6  HTML 4.0</td>
</tr>
<tr>
<td></td>
<td>7  Printer</td>
</tr>
<tr>
<td></td>
<td>8  Rich Text Format (RTF)</td>
</tr>
<tr>
<td></td>
<td>9  Word for Windows (DOC)</td>
</tr>
<tr>
<td></td>
<td>10  XML</td>
</tr>
<tr>
<td></td>
<td>11  Text</td>
</tr>
<tr>
<td>-o</td>
<td>Output file name. If this is omitted, RunReport uses the format <code>HomeDir\Report\OutputFiles\UserName\TaskName_Site_TaskNum.Extension</code>.</td>
</tr>
<tr>
<td>-n</td>
<td>Number of copies to print, if the report output is sent directly to a printer (output format 7). Default value is 1.</td>
</tr>
</tbody>
</table>
| -e      | Error file name. If a file name without a path is passed in, RunReport adds `HomeDir\Report\Errors\UserName\`.
|         | If this parameter is omitted, RunReport uses the format `HomeDir\Report\Errors\UserName\TaskName_TaskNum.txt`.
|         | Infor ERP TaskMan puts the contents of this file in the error message field on the Background Task History form and then deletes the file. |
| -font   | Name of the font to use for the report. The font (Arial, Times New Roman, etc.) must be loaded on the machine where RunReport is running. This value defaults to the font used when the report was created. If that font is not on the RunReport machine, Crystal Reports defaults the value to Arial. RunReport loops through all of a report’s fields and resets the font. If there are fields where the font should never be reset (for example, barcode fields), then use the field prefix LEAVEFONT or LF_ (see page 32). |
When TaskMan and MGCore generate reports, they format and then execute command lines like the following (replace SyteLine_Forms with the appropriate database name).

If TaskMan is run with the debug startup parameter, this command line is printed to the Application Event log. Reports can be tested by putting this command line in a batch file and executing it. The **** -p parameter should be replaced by the appropriate password.
Example Using ODBC (RDO) or OLE DB (ADO) ODBC Connection

RunReport.exe -d MyServerDSN -l sa -p **** -s SyteLine_Forms.dbo.Strings -f 1 -r ItemQuantitiesbyABCCode -t 199 -u sa -e "D:\TaskMan\Report\Errors\sa\ItemQuantitiesbyABCCodeReport_MI_199.txt" -site "MI" -db SLMI_App -n 1 -m 
-LIT-(A)-LIT-(B)-LIT-(C),-LIT-(M)-LIT-(P)-LIT-(T),-LIT-(M)-LIT-(T)-LIT-(F)-LIT-(O),-LIT-(1),-LIT-(1),-LIT-(),-LIT-(DIS1),-LIT-(MAIN),-LIT-(AD-10000),-LIT-(AL-10000),-LIT-(),-LIT-(),-LIT-()

Example Using OLE DB (ADO) Direct SQL Connection

RunReport.exe -svr MyServer -l sa -p **** -s SyteLine_Forms.dbo.Strings -f 1 -r ItemQuantitiesbyABCCode -t 199 -u sa -e "D:\TaskMan\Report\Errors\sa\ItemQuantitiesbyABCCodeReport_MI_199.txt" -site "MI" -db SLMI_App -n 1 -m 
-LIT-(A)-LIT-(B)-LIT-(C),-LIT-(M)-LIT-(P)-LIT-(T),-LIT-(M)-LIT-(T)-LIT-(F)-LIT-(O),-LIT-(1),-LIT-(1),-LIT-(),-LIT-(DIS1),-LIT-(MAIN),-LIT-(AD-10000),-LIT-(AL-10000),-LIT-(),-LIT-(),-LIT-()

Substitution Keywords

RunReport also support severals keywords. These keywords are replaced with values if they occur in the text of formulas used in reports.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BG<del>USERID</del></td>
<td>User who submitted the report.</td>
</tr>
<tr>
<td>BG<del>UID</del></td>
<td>User who submitted the report. This parameter is replaced when passed from a form, as well as being replaced in report formulas.</td>
</tr>
<tr>
<td>BG~SITEID</td>
<td>Site field from the parms table.</td>
</tr>
<tr>
<td>BG<del>COMPANYNAME</del></td>
<td>Company field from the parms table.</td>
</tr>
</tbody>
</table>

Debug Mode

RunReport can be run in debug mode. To set this up, use TaskMan start parameter `debugrep` (see page 49). The output from debug mode is explained in "RunReport Debug Messages" on page 82.
SyteLine Stored Procedure Example

The following script creates a sample stored procedure (Rpt_ReportDemoSp) that is used as a reference for examples in chapters 2 and 3. To run this script:

1. Open SQL Query Analyzer.
2. Select Query>Change Database and point to a test SyteLine application database.
3. Copy all of the code below into Query Analyzer.
4. Execute the script. This places the SP in the "Stored Procedures" area of the application database you selected.
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Rpt_ReportDemoSp]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Rpt_ReportDemoSp]
GO
CREATE  PROCEDURE Rpt_ReportDemoSp
(  @OrderStarting   CoNumType = NULL
 , @OrderEnding     CoNumType = NULL
 , @OrderDateStarting   DateType = NULL
 , @OrderDateEnding     DateType = NULL
 , @OrderDateStartingOffset DateOffsetType = NULL
 , @OrderDateEndingOffset DateOffsetType = NULL
 , @DisplayHeader   FlagNyType = NULL
 , @ShowInternal    FlagNyType = NULL
 , @ShowExternal    FlagNyType = NULL
 ) AS
-- Transaction management is not provided by Crystal, so a transaction
-- is started here.
BEGIN TRANSACTION
SET XACT_ABORT ON
-- Set the isolation level specified for the background task or use
-- the system default.
IF dbo.GetIsolationLevel(N'ReportDemo') = N'COMMITTED'
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED
ELSE
    SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
-- A session context is created so session variables can be used.
DECLARE @RptSessionID RowPointerType
EXEC InitSessionContextSp
    @ContextName = 'Rpt_ReportDemoSp'
 , @SessionID   = @RptSessionID OUTPUT
-- replace NULL input parameters with Min or Max values
SET @OrderStarting = ISNULL(dbo.ExpandKyByType('CoNumType', @OrderStarting),
    dbo.LowString('CoNumType'))
SET @OrderEnding = ISNULL(dbo.ExpandKyByType('CoNumType', @OrderEnding),
    dbo.HighString('CoNumType'))
EXEC ApplyDateOffsetSp @Date = @OrderDateStarting OUT, @Offset =
    @OrderDateStartingOffset, @IsEndDate = 0
EXEC ApplyDateOffsetSp @Date = @OrderDateEnding OUT, @Offset =
    @OrderDateEndingOffset, @IsEndDate = 1
IF @DisplayHeader IS NULL SET @DisplayHeader = 1
IF @ShowInternal IS NULL SET @ShowInternal = 1
IF @ShowExternal IS NULL SET @ShowExternal = 1
-- Create a temp table for the report output
DECLARE
    @Cust    AS CustNumType
 , @Address AS LongAddress
 , @Status  AS CoStatusType
 , @CoLine  AS CoLineType
 , @Item    AS ItemType
 , @Desc    AS DescriptionType
 , @Price   AS AmountType
SELECT
    @OrderStarting     AS CoNum
 , @CoLine            AS Line
 , @Cust              AS CustNum
 , @OrderDateStarting AS OrderDate

**, @Status** AS Status,
*, @Address** AS Address,
*, @Item** AS Item,
*, @Desc** AS Description,
*, @Price** AS NetPrice,
*, @RptSessionID** AS RowPointer,
*, @ShowInternal** AS NoteExistsFlag
INTO #ReportOutput
WHERE 1=0
-- get the report output
INSERT INTO #ReportOutput
SELECT
  co.co_num,
  coi.co_line,
  co.cust_num,
  co.order_date,
  co.stat,
  LEFT(CASE WHEN co.cust_num <> '' AND co.cust_num IS NOT NULL THEN
dbo.FormatAddress(co.cust_num, 0) ELSE '' END, 255),
  coi.item,
  it.description,
  ROUND((1 - (ISNULL(coi.disc,0)) / 100) * (ISNULL(coi.qty_ordered_conv,0)) * (ISNULL(coi.price_conv,0)),cur.places),
  co.RowPointer,
  co.NoteExistsFlag
FROM co co
LEFT OUTER JOIN coitem coi ON co.co_num = coi.co_num
LEFT OUTER JOIN Item_all it ON it.item = coi.item AND it.site_ref = coi.ship_site
LEFT OUTER JOIN custaddr adr ON adr.cust_num = co.cust_num AND adr.cust_seq = co.cust_seq
LEFT OUTER JOIN currency cur ON cur.curr_code = adr.curr_code
WHERE
  co.co_num >= @OrderStarting AND co.co_num <= @OrderEnding
AND co.order_date >= @OrderDateStarting AND co.order_date <= @OrderDateEnding
-- return report record set
SELECT
  CoNum,
  Line,
  CustNum,
  OrderDate,
  Status,
  Item,
  Description,
  NetPrice,
  NoteExists = dbo.ReportNotesExist('co', RowPointer, @ShowInternal, @ShowExternal, NoteExistsFlag)
FROM #ReportOutput
ORDER BY CoNum, Line
COMMIT TRANSACTION
EXEC CloseSessionContextSp @SessionID = @RptSessionID
GO
Symbols
~LIT~ Keywords, Viewing in Query Analyzer 86
~LIT~ Syntax 46, 48
   in RunReport Command Line 92
A
Active Background Tasks Form 9, 40
ActiveBGTasks 17, 47
ActiveX Component 69
Adding Tasks 40
ApplyDateOffsetSp 24
B
Background Queue 37, 40
   Submitting Tasks to 39
Background Queue Form 8
Background Task Definitions Form 7, 35
Background Task History Form 9, 40, 55
Background Task, Adding 35
BackGroundQueueDeleteSP 18
BackGroundQueueSP 18, 40
Barcodes
   Adding 27
   Type Fonts 19
BG~ Keywords 47
BGTaskDefinition 47
BGTaskDefinitions 17, 18
BGTaskHistory 17, 47
BGTaskName Variable 37
BGTaskParms Variable 37
BGTaskSubmit 18, 39
BGTaskSubmitSp 40
Button, Adding to Form 37
C
Cap, Record Collection 20
Company, RunReport/RunCrystal Substitution Keyword
93
Crystal Reports, see Reports
Current Session Values 23
Customer Document Profile 20
Customization 18
D
Database Name
   RunReport Command 91
Database Name, Substitution Keyword 47
Database Name, Verifying 25
Database Name, Wrong One Used 87
Database Number, Substitution Keyword 47
Date Format
   Language IDs Form 19
   Preventing Overrides 32
   RunReport Command 92
Date Offset Fields, Adding to Form 37
Date Offset Parameters in Stored Procedures 24
Debug Messages (RunReport) 82
Debug Mode Messages (TaskMan) 73
Debug Mode, Running Infor ERP Task Manager in 49
Debug Parameters, in Task Manager 49
download TaskMan Startup Parameter 82
Default Printer 56
DSN
   RunReport Command 90
E
E-mail 21, 43
E-mail Notification Not Received 57
Error 128 - Error Running Report 67
Error 500 - Not Enough Memory for Operation 68
Error 515 - Error Starting Report 68
Error 534 - Error Detected by Database DLL 68
Error 534 - Error detected by Database DLL 25
Error File
   Substitution Keyword 47
   Substitution Keyword to Prevent Deletion 48
Error File Name
   RunReport Command 91
Error Log Location 42
Error Message Locations 55
Errors, see Troubleshooting
Event Handlers 37
Event Log 50
Event Messages from Infor ERP Task Manager 70
Event Viewer 55
Excluded Tasks Form 8
EXE Files
   Database Connections 45
   Returning Error Information through Infor ERP Task Manager 45
EXE Files, Running Through Infor ERP Task Manager 45
Executable Name 7, 35
Executable Type 7, 35
ExpandKyByType 23
Export Formats and DLLs 63, 66
F
Fax Header, Substitution Keywords 47, 48
Field Name is not Known Error 67
Field, Adding for Report Parameters 36
File Not Found Error 59
Font Descriptor 18
Font Name
   RunReport Command 91
Fonts Used in Reports 18
Forms
   Adding Date Offset Fields 37
Adding Notes 30
Adding Report Preview 39
Modifying to Run Report 36
Submitting Task to BG Queue 39
Formula, Maximum Length 68

G
General Parameters Form 43, 62
Group
   RunReport Command 91

H
Help Files, Crystal Reports 7
HighDate 23
HighInt 23
HighString 23
Home Directory
   RunReport Command 90
Home Directory SubstitionKeyword 47

I
IDO Methods, Running through Infor ERP Task Manager 46
Infor ERP Task Manager
   Configuring Additional DSNs 42
   Debug Mode 49
   Debug Mode Messages 73
   Does Not Start (Error) 64
   Error and Output File Names 42
   Event Log 50
   Event Messages 70
   Executing Stored Procedures 44
   Installation 42
   Keywords 37, 40, 44, 47
   Nowait Keyword 49
   Options 52
   Overview 41
   Running Executables 45
   Running IDO Methods 46
   Setting Up as ODBC DSN 42
   Setting Up as Service 42
   Standard Installation 42
Initial Catalog 87
Input Parameters
   Minimum and Maximum Values 23
   Stored Procedure 23
Installation
   Development Test Environment 43
   Standard Environment 42
Intranets Form 59, 64
Invalid String Table Name 63
IsInternalNote 29
Isolation Level, Substitution Keyword 36, 47

K
Keywords, see Infor ERP Task Manager Keywords

L
Labels
   Using Formulas 25
   Labels Not Displaying Properly Error 62
   Labels Not Replaced with String Table Values Error 43
   Language user 20
   Language Code 20
   Language IDs Form 19
   Languages Used in Reports 19, 33
   Launching a Report 37
   List Pair Display Values, in RPT Files 26
   Login
      RunReport Command 90
   Logos, Modifying Report Logos 31
   LowDate 23
   LowInt 23
   LowString 23

M
Max Concurrent 7, 35, 65
Memory Error 68
Messages, Translating 33
MGCore.dll 17, 43
IDO 40
Location 61
MGReportProcessor.dll 17
Microsoft Event Viewer 50, 55, 70
Missing or Out-of-Date Export DLL Error 66
Missing RPT File Error 43, 59
Modification 18
Multi Lingual Order Invoice Form 20
Multi Lingual Purchase Orders Form 20
Multi-Site Output File Names 42

N
No Database Definitions Defined (Error) 64
No Default Set for Parameter 38 (Error) 66
No Output Error 64
NoteHeaders 29
Notes
   Adding as Subreport 29
   Adding through a Form 30
   Adding to Reports 29
   Nowait Keyword for Task Manager 49
   Standard Installation 42
   Number of Copies
      RunReport Command 91
   Numeric Format
      Language IDs Form 19
      Preventing Overrides 32
      RunReport Command 92

O
ObjectNotes 29
ODBC DSN
   Substitution Keyword 47
Output Directory Path, Substitution Keyword 47
Output File
   RunReport Command 91
Output File Location 42
Output Format 9
   RunReport Command 91

P
Password
   RunReport Command 90
Preview Report
   Error Log 55
Preview Reports
   Adding to Forms 39
   Errors 59
Print Preview
   Languages 20
Printer Name
   RunReport Command 92
Process Defaults 49, 52
ProcessErrorLog 45

Q
Query Engine Error 67

R
RaiseErrorSp 61
Read Committed 36
Read Uncommitted 36
RefRowPointer 29
Remaining Text Not Part of Formula 67
Report Criteria Form 36
Report Design Component 41
Report Designer Component 89
Report Name
   RunReport Command 90
Report Options Form 9, 56, 57
Report Parameters, Viewing in BG Task History 86
ReportNotesExist 29
ReportNotesView 29
ReportOptions 17

Reports
   Adding Background Task 35
   Adding Button to Form 37
   Adding Notes 29
   Adding Notes as Subreport 29
   Adding Notes through a Form 30
   Adding Parameter Fields to Form 36
   Adding Preview 39
   Date Offset 37
   Error and Output File Names 42
   Error Log 42
   Help Files 7
   Launching from Form 37
   Linking to SyteLine 35
   Modifying a Form 36
   Modifying Report Logos 31
   Output File Location 42
   RPT File Location 42
   Strings Table to Use 42
   Submitting Task to BG Queue 39
Reports Not Printing 56
Result Set, in Stored Procedures 23
RPT Files
   Creating from Stored Procedures 25
   Formulas for Labels 25
   List Pairs 26
   Location 42
   Storage of 25
Rpt_ReportDemoSp Example Stored Procedure 95
RunCrystal.exe
   Placement on Server 42
   Substitution Keywords 93
RunReport.exe 17, 65
   Command Line Switches 90
   Running from the Command Line 90
   Substitution Keywords 93

S
Scheduling Information, Checking through SQL Enterprise Manager 40
Server, Substitution Keyword 47
Session, RunReport Command 91
Setting in Stored Procedure 23
Simple MAPI Protocol 43
Site
   RunReport Command 92
   Site ID, RunReport/RunCrystal Substitution Keyword 93
Sites/Entities Form 9, 11, 42, 43, 62
SMTP Protocol 43
SpecificNotes 29
SQL Login, Substitution Keyword 47
SQL Profiler 41, 51, 56
Stored Procedures
   Date Offset Parameters 24
   Example 23, 95
   Executing Through Infor ERP Task Manager 44
   Executing through Infor ERP Task Manager 41
   Input Parameters 23
   Result Set 23
   Structure 23
   Temp Tables 23
   User Defined Types 23
String Translation
   RunReport Debug 86
String Translation in RPT Files 25
Strings Table 9, 11, 20, 25, 42, 43
   Errors 62, 63
   RunReport Command 91
   Used with Reports 42
Substitution Keywords 47
   RunCrystal and RunReport 93
   Substitution Keywords, see Infor ERP Task Manager Keywords
   SystemNotes 29
SyteLineLabelStringIDs 27

T
Task Man Path 43, 59
Task Manager, see Infor ERP Task Manager
Task Name 35
  Substitution Keyword 47
Task Number
  RunReport Command 90
  Substitution Keyword 47
Taskman.exe 17
  Placement on Server 42
TASKNUMBER Input Parameter 40
Tasks, Splitting Up 21
TASKSTATUS Input Parameter 39
Temp Tables, in Stored Procedures 23
This Field Name Is Not Known (Error) 68
TMMsgs.dll 17, 65
Translating Messages 33
Troubleshooting 55
  Batches Fail 65
  E-mail Notification Not Received 57
  Error 128 67
  Error 500 68
  Error 515 68
  Error 534 68
  Field Name is not Known 67
  Field Name Not Known 68
  File Not Found 59
  Labels not Displaying Properly 62
  Missing or Out-of-Date Export DLL 66
  Missing RPT File 59
  No Default Set for Parameter 38 66
  No Output 64
  Notes Not Printing on Report 65
  Query Engine Error 67
  Remaining Text Not Part of Formula 67
  Report Preview 59
  Reports Not Printing 56
  Task Manager Does Not Start 64
  Task Runs - No History Record 66
  Truncated Data in Text Output 59
  T-SQL Error, Getting More Meaningful Message 60

U
User Defined Types 23
User ID
  RunReport Command 90
  RunReport/RunCrystal Substitution Keyword 93
  Substitution Keyword 47
Users Form 57

V
Vendor Document Profile 20
Verify Database 59
Verifying the Database Name 25

W
WinStudioAppIdEditor 61